AD NUMBER

AD871261

NEW LIMITATION CHANGE

TO

Approved for public release, distribution unlimited

FROM

Distribution authorized to U.S. Gov't. agencies and their contractors; Critical Technology; NOV 1969. Other requests shall be referred to Naval Training Deveice Center, Orlando, FL.

AUTHORITY

USNTDC ltr, 9 Dec 1971

15

Technical Report: NAVTRADEVCEN 68-C-0050-2

ADVANCED SUBMARINE SYSTEMS PROGRAMING

(Final)

Brenton R. Groves, Ph. D.
James T. Dorsey
Dennis Tucker

Goodyear Aerospace Corporation
Akron, Ohio

Contract N61339-68-C-0050

November 1969

<u>DoD Distribution Statement</u>

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of the Commanding Officer, Naval Training Device Center, (423), Orlando, Florida 32813

NAVAL TRAINING DEVICE CENTER

ORLANDO, FLORIDA

382

Best
Available
Copy

Technical Report: NAVTRADEVCEN 68-C-0050-2

ADVANCED SUBMARINE SYSTEMS PROGRAMING

ABSTRACT

This programing report is the result of a study leading to the determination of the optimum sets of equations of motion to be used with two general types of submarine control trainers. The starting point was the Naval Ship Research and Development Center standard equations of motion for submarine simulators.

Two complete submarine simulation programs using these equations are given; one for six-degrees-of-freedom and one for the longitudinal three-degrees-of-freedom. A number of programs are included to assist the researcher in interpreting the results of the main simulation program. A compact submarine simulation program for use with a small computer is given and a method of generating random ocean wave amplitudes is outlined along with its program.

This report describes the programs, including listing in FORTRAN, flow charts, input decks, and typical output sheets, but does not tell how they are to be used. It should be used in conjunction with Advanced Submarine Systems Equation Study, NAVTRADEVCEN 68-C-0050-1 which describes the work performed under this study.

FOREWORD

This report presents computer programs which allow various investigations of submarine simulation. Descriptions, flow charts, and listings are presented for each program. Important uses of the programs include coefficient reduction of any class of submarine, checking accuracy of coefficients when operational data is available, and research in casualty situations.

NAVTRADEVCEN 68-C-0050-1 gives an overall description of the equations study. NAVTRADEVCEN 68-C-0050-3 presents results of the computer programs using the SS(N)594 submarine as the demonstration model.

Charles A. Rumbough
CHARLES A. RUMBOUGH
Project Engineer

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

SECTION I

INTRODUCTION

This report is the result of a study leading to the determination of the optimum sets of equations of motion, to be used with two general types of submarine control trainers.

Two complete submarine simulation programs are given; one for six-degrees-of-freedom and one for three-degrees-of-freedom. A number of programs are included to assist the researcher in interpreting the results of the main simulation program. A compact program for use with a small computer is also included.

This report describes the programs, including listing in FORTRAN, but does not tell how they are to be used. It should be used in conjunction with Advanced Submarine System Equation Study, NAVTRADEVCEN 68-C-0050-1 which describes the work performed under this study.

SECTION II

PROGRAM DESCRIPTIONS

A. PROGRAM EB920, SUBMARINE SIMULATION

1. DESCRIPTION

This program calculates the dynamic changes of a body's position and attitude as a function of time. The vehicle is a submarine in this case but any vehicle can be simulated if the coefficients of the equations of motion are known. The equations used for the mathematical model are developed in "Standard Equations of Motion for Submarine Simulation", Report 2510[1*] by Morton Gertler and Grant R. Hagen of the Naval Ship Research and Development Center in Washington, D.C. They are designated as the NSRDC Standard Equations and the terminology in this program follows this report.

The NSRDC Standard Equations cover all phases of submarine motion simulation in six-degrees-of-freedom including emergency recoveries after casualties. NSRDC Report 2510 contains a brief history, defines the mathematical model, discusses the coefficients required, and sets a standard to be used in the simulation of submarines. Equations (1) through (6) present the equations of motion in the following order: axial force, lateral force, normal force, rolling moment, pitching moment, and yawing moment. In addition certain Kinematic relations are given in equation (7).

The program is written in basic FORTRAN IV for use on any digital computer with a FORTRAN compiler. It occupied 20K words when run on an IBM 360/40 computer. Program results are printed as a function of time and individual parameters can be plotted if associated CALCOMP plotter software is available.

Figure 1 is a block diagram showing the general outline of the program and the subroutines used. A number of different options are available to the program user. They are listed below.

a. Programmed control surface and thrust values.

1. Climbing turns - fixed elevator and rudder deflection; for climb or turn (without autopilot):

2. Meander or overshoot

3. Modified climbing turns - surfaces deflected at controlled rates to specified values.

4. Flat turn (with autopilot)

5. Climbing impulse

* Superscript numbers indicate references

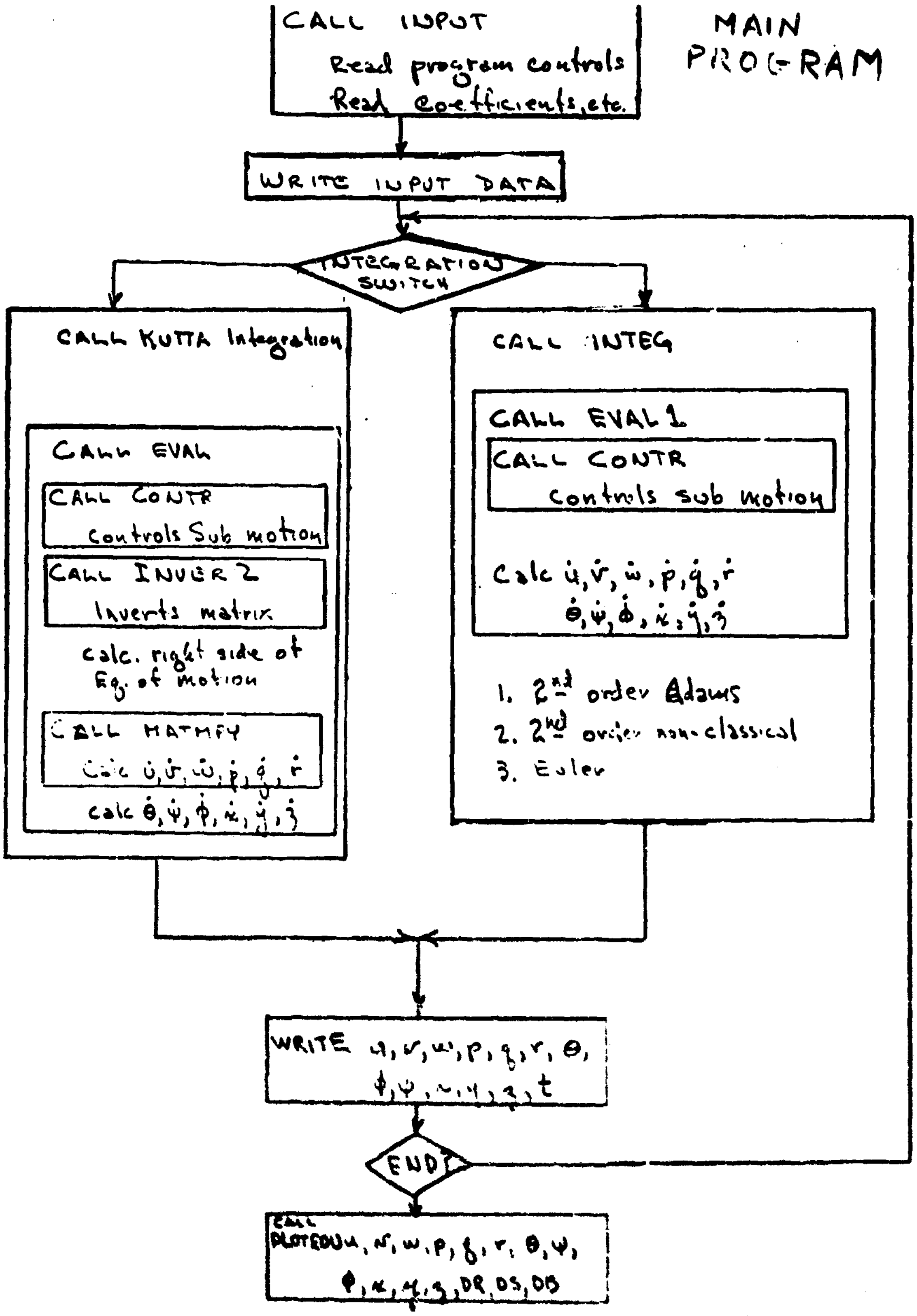


Figure 1. Research Simulation Program Block Diagram

AXIAL FORCE (1)

$$m\left[\dot{u} - vr + wq - x_G (q^2 + r^2) + y_G (pq - \dot{r}) + z_G (pr + \dot{q})\right] =$$

$$+ \frac{\rho}{2} \ell^4 \left[X_{qq}' q^2 + X_{rr}' r^2 + X_{rp}' rp\right]$$

$$+ \frac{\rho}{2} \ell^3 \left[X_{\dot{u}}' \dot{u} + X_{vr}' vr + X_{wq}' wq\right]$$

$$+ \frac{\rho}{2} \ell^2 \left[X_{uu}' u^2 + X_{vv}' v^2 + X_{ww}' w^2\right]$$

$$+ \frac{\rho}{2} \ell^2 u^2 \left[X_{\delta r\, \delta r}' \delta r^2 + X_{\delta s\, \delta s}' \delta s^2 + X_{\delta b\, \delta b}' \delta b^2\right]$$

$$+ \frac{1}{2} \rho \ell^2 \left[a_i u^2 + b_i uu_c + c_i u_c^2\right]$$

$$- (W - B) \sin \theta$$

$$+ \frac{\rho}{2} \ell^2 \left[X_{vv\eta}' v^2 + X_{ww\eta}' w^2 + X_{\delta r \delta r \eta}' \delta_r^2 u^2\right.$$

$$\left. + X_{\delta s \delta s \eta} \delta_s^2 u^2 \right] (\eta - 1)$$

LATERAL FORCE

$$m\left[\dot{v} - wp + ur - y_G(r^2 + p^2) + z_G(qr - \dot{p}) + x_G(qp + \dot{r})\right] =$$

$$+\frac{\rho}{2}\ell^4\left[Y_{\dot{r}}'\dot{r} + Y_{\dot{p}}'\dot{p} + Y_{p|p|}'p|p| + Y_{pq}'pq + Y_{qr}'qr\right]$$

$$+\frac{\rho}{2}\ell^3\left[Y_{\dot{v}}'\dot{v} + Y_{vq}'vq + Y_{wp}'wp + Y_{wr}'wr\right]$$

$$+\frac{\rho}{2}\ell^3\left[Y_r'ur + Y_p'up + Y_{|r|\delta r}'u|r|\delta r + Y_{v|r|}'\frac{v}{|v|}\left|(v^2 + w^2)^{\frac{1}{2}}\right||r|\right]$$

$$+\frac{\rho}{2}\ell^2\left[Y_*'u^2 + Y_v'uv + Y_{v|v|}'v\left|(v^2 + w^2)^{\frac{1}{2}}\right|\right]$$

$$+\frac{\rho}{2}\ell^2\left[Y_{vw}'vw + Y_{\delta r}'u^2\,\delta r\right]$$

$$+ (W - B)\cos\theta\sin\phi$$

$$+\frac{\rho}{2}\ell^3 Y_{r\eta}'ur(\eta-1)$$

$$+\frac{\rho}{2}\ell^2\left[Y_{v\eta}'uv + Y_{v|v|\eta}'v\left|(V^2 + w^2)^{\frac{1}{2}}\right| + Y_{\delta r\eta}'\delta_r u^2\right](\eta-1)$$

NORMAL FORCE

$$m\left[\dot{w} - uq + vp - z_G (p^2 + q^2) + x_G (rp - \dot{q}) + y_G (rq + \dot{p})\right] =$$

$$+\frac{\rho}{2} \ell^4 \left[Z_{\dot{q}}' \dot{q} + Z_{pp}' p^2 + Z_{rr}' r^2 + Z_{rp}' rp\right]$$

$$+\frac{\rho}{2} \ell^3 \left[Z_{\dot{w}}' \dot{w} + Z_{vr}' vr + Z_{vp}' vp\right]$$

$$+\frac{\rho}{2} \ell^3 \left[Z_q' uq + Z_{|q|\delta s}' u|q|\delta s + Z_{w|q|}' \frac{w}{|w|}\left|(v^2 + w^2)^{\frac{1}{2}}\right| |q|\right]$$

$$+\frac{\rho}{2} \ell^2 \left[Z_*' u^2 + Z_w' uw + Z_{w|w|}' w\left|(v^2 + w^2)^{\frac{1}{2}}\right|\right]$$

$$+\frac{\rho}{2} \ell^2 \left[Z_{|w|}' u|w| + Z_{ww}' \left|w (v^2 + w^2)^{\frac{1}{2}}\right|\right]$$

$$+\frac{\rho}{2} \ell^2 \left[Z_{vv}' v^2 + Z_{\delta s}' u^2 \delta s + Z_{\delta b}' u^2 \delta b\right]$$

$$+ (W - B) \cos\theta \cos\phi$$

$$+\frac{\rho}{2} \ell^3 Z_{q\eta}' uq (\eta - 1)$$

$$+\frac{\rho}{2} \ell^2 \left[Z_{w\eta}' uw + Z_{w|w|\eta}' w\left|(v^2 + w^2)^{\frac{1}{2}}\right| + Z_{\delta s\eta}' \delta_s u^2\right](\eta - 1)$$

ROLLING MOMENT

$$I_x\dot{p} + (I_z - I_y)\,qr - (\dot{r} + pq)\,I_{xz} + (r^2 - q^2)\,I_{yz} + (pr - \dot{q})\,I_{xy}$$

$$+ m\left[y_G(\dot{w} - uq + vp) - z_G(\dot{v} - wp + ur)\right] =$$

$$+\frac{\rho}{2}\,\ell^5\left[K_{\dot{p}}'\,\dot{p} + K_{\dot{r}}'\,\dot{r} + K_{qr}'\,qr + K_{pq}'\,pq + K_{p|p|}'\,p|p|\right]$$

$$+\frac{\rho}{2}\,\ell^4\left[K_p'\,up + K_r'\,ur + K_{\dot{v}}'\,\dot{v}\right]$$

$$+\frac{\rho}{2}\,\ell^4\left[K_{vq}'\,vq + K_{wp}'\,wp + K_{wr}'\,wr\right]$$

$$+\frac{\rho}{2}\,\ell^3\left[K_*'\,u^2 + K_v'\,uv + K_{v|v|}'\,v\left|(v^2 + w^2)^{\frac{1}{2}}\right|\right]$$

$$+\frac{\rho}{2}\,\ell^3\left[K_{vw}'\,vw + K_{\delta r}'\,u^2\,\delta r\right]$$

$$+ (y_G W - y_B B)\cos\theta\cos\phi - (z_G W - z_B B)\cos\theta\sin\phi \,\Big]$$

$$+\frac{\rho}{2}\,\ell^3\,K_{*\eta}'\,u^2\,(\eta - 1)$$

PITCHING MOMENT

$$I_y \dot{q} + (I_x - I_z)\, rp - (\dot{p} + qr)\, I_{xy} + (p^2 - r^2)\, I_{zx} + (qp - \dot{r})\, I_{yz}$$

$$+ m \left[z_G (\dot{u} - vr + wq) - x_G (\dot{w} - uq + vp) \right] =$$

$$+ \frac{\rho}{2} \ell^5 \left[M_{\dot{q}}' \dot{q} + M_{pp}' p^2 + M_{rr}' r^2 + M_{rp}' rp + M_{q|q|}' q|q| \right]$$

$$+ \frac{\rho}{2} \ell^4 \left[M_{\dot{w}}' \dot{w} + M_{vr}' vr + M_{vp}' vp \right]$$

$$+ \frac{\rho}{2} \ell^4 \left[M_q' uq + M_{|q|\delta s}' u|q|\delta s + M_{|w|q}' |(v^2 + w^2)^{\frac{1}{2}}| q \right]$$

$$+ \frac{\rho}{2} \ell^3 \left[M_*' u^2 + M_w' uw + M_{w|w|}' w |(v^2 + w^2)^{\frac{1}{2}}| \right]$$

$$+ \frac{\rho}{2} \ell^3 \left[M_{|w|}' u|w| + M_{ww}' |w (v^2 + w^2)^{\frac{1}{2}}| \right]$$

$$+ \frac{\rho}{2} \ell^3 \left[M_{vv}' v^2 + M_{\delta s}' u^2 \delta s + M_{\delta b}' u^2 \delta b \right]$$

$$- (x_G W - x_B B) \cos\theta \cos\phi - (z_G W - z_B B) \sin\theta$$

$$+ \frac{\rho}{2} \ell^4 M_{q\eta}' uq (\eta - 1)$$

$$+ \frac{\rho}{2} \ell^3 \left[M_{w\eta}' uw + M_{w|w|\eta}' w |(v^2 + w^2)^{\frac{1}{2}}| + M_{\delta s\eta}' \delta_s u^2 \right] (\eta - 1)$$

YAWING MOMENT

$$I_z \dot{r} + (I_y - I_x)\, pq - (\dot{q} + rp)\, I_{yz} + (q^2 - p^2)\, I_{xy} + (rq - \dot{p})\, I_{zx}$$

$$+ m \left[x_G (\dot{v} - wp + ur) - y_G (\dot{u} - vr + wq) \right] =$$

$$+ \frac{\rho}{2} \ell^5 \left[N_{\dot{r}}' \dot{r} + N_{\dot{p}}' \dot{p} + N_{pq}' pq + N_{qr}' qr + N_{r|r|}' r|r| \right]$$

$$+ \frac{\rho}{2} \ell^4 \left[N_{\dot{v}}' \dot{v} + N_{wr}' wr + N_{wp}' wp + N_{vq}' vq \right]$$

$$+ \frac{\rho}{2} \ell^4 \left[N_p' up + N_r' ur + N_{|r|\delta r}' u|r|\delta r + N_{|v|r}' \left|(v^2 + w^2)^{\frac{1}{2}}\right| r \right]$$

$$+ \frac{\rho}{2} \ell^3 \left[N_*' u^2 + N_v' uv + N_{v|v|}' v \left|(v^2 + w^2)^{\frac{1}{2}}\right| \right]$$

$$+ \frac{\rho}{2} \ell^3 \left[N_{vw}' vw + N_{\delta r}' u^2 \delta r \right]$$

$$+ (x_G W - x_B B) \cos\theta \sin\phi + (y_G W - y_B B) \sin\theta$$

$$+ \frac{\rho}{2} \ell^4 N_{r\eta}' ur (\eta - 1)$$

$$+ \frac{\rho}{2} \ell^3 \left[N_{v\eta}' uv + N_{v|v|\eta}' v \left|(v^2 + w^2)^{\frac{1}{2}}\right| + N_{\delta r\eta}' \delta_r u^2 \right] (\eta - 1)$$

KINEMATIC RELATIONS (7)

$$V^2 = u^2 + v^2 + w^2$$

$$\dot{\theta} = q\cos\phi - r\sin\phi$$

$$\dot{\psi} = \frac{q\sin\phi + r\cos\phi}{\cos\theta}$$

$$\dot{\phi} = p + \dot{\psi}\sin\phi$$

$$\dot{x}_0 = u\cos\theta\cos\psi + v(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi)$$
$$+ w(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)$$

$$\dot{y}_0 = u\cos\theta\sin\psi + v(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi)$$
$$+ w(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)$$

$$\dot{z}_0 = -u\sin\theta + v\cos\theta\sin\phi + w\cos\theta\cos\phi$$

6. Turning impulse (with autopilot)

7. Acceleration/deceleration (with autopilot)

8. Maximum acceleration/deceleration (with autopilot)

b. Integration Methods

1. Fourth Order Runge-Kutta

2. Second order Adams

3. Second order non-classical

4. Euler

c. Variable integration step size (H, or integration time increment). This option allows study of the allowable or optimum integration step size, H, as a function of output variable accuracy.

d. Initial Conditions

The program will accept values of initial conditions for control surface positions, speed, attitude and depth. This allows the submarine to be placed in a steady state dive, level flight, etc. at time zero or start of the computer run. These inputs are:

UC - command speed
DR - rudder position
DS - sternplane position
DB - sailplane position
W - component of velocity in the z-direction
Q - angular acceleration component about the y-axis
THETA - angle of pitch
Z - depth

All other parameters of motion, v, p, r, ψ, ϕ, x, and y, can also be set as initial conditions. These parameters are usually zero at time zero. The values for steady-state level flight can be calculated with program EC470.

e. Output Options

Any time interval can be set between printing out the parameters of motion. If a CALCOMP plotter is available, any parameter and the control surfaces can be plotted as a function of time in any order.

2. SUBROUTINE DESCRIPTIONS

The subroutines utilized by this program are shown in block diagram

form in figure 1. A brief description of each subroutine is included for clarification of operation of the total program.

a. INPUT - This subroutine reads a data deck specifying program options, controls and initial dynamic conditions, and the particular coefficients and constants for the submarine to be simulated. It provides for holding initial values of all variables for use on a subsequent run. It provides for reading additional cards for successive runs, and finally for exit from the program on a blank card.

b. KUTTA - This is an integration subroutine that uses the Runge-Kutta, 4th order integration method to integrate the equations of motion over the time period required in accordance with the equation:

$$Y_{n+1}(I) = Y_n(I) + 1/6\,(K_o(I) + 2K_1(I) + 2K_2(I) + K_3(I))$$

where $K_o(I) = h \cdot \mathrm{EVAL}\,(Y_n(I))$

$K_1(I) = h \cdot \mathrm{EVAL}\,(Y_n(I) + \frac{1}{2} K_o(I))$

$K_2(I) = h \cdot \mathrm{EVAL}\,(Y_n(I) + \frac{1}{2} K_1(I))$

$K_3(I) = h \cdot \mathrm{EVAL}\,(Y_n(I) + K_2(I))$

I = 1 to 12

h = integration time interval

$Y_n(I)$ = motion parameter u, v, etc. at the nth cycle

EVAL = equation of motion subroutine

This subroutine takes a twelve-element matrix from subroutine EVAL corresponding to accelerations and velocities; integrates over the interval, h; and returns twelve new velocities or positions. (Corresponding to u, v, w, p, q, r, Θ, Ψ, ϕ, x, y, z). I runs from 1 to 12 to cover all the terms to be integrated. The EVAL subroutine is entered four times in order to calculate the four K's since each one is dependent on the last one. The integration method is started by zeroing the working storage during the first pass through the subroutine.

The mathematical reasoning behind this particular algorithm is given in F.B. Hildebrand, "Introduction to Numerical Analysis", McGraw-Hill 1956, page 237.

c. CONTR - This subroutine allows various submarine control manuevers to be selected that are used in submarine research studies. The subroutine varies the control surfaces during the run to allow meander, submerged turns, overshoot, acceleration, etc. to be made. The block diagram of figure 2 outlines the control subroutine. In all controls, the variable

SUBROUTINE CONTR

GO TO(1001, 1001, 1003, 1004, 1005, 1006, 1007), NS

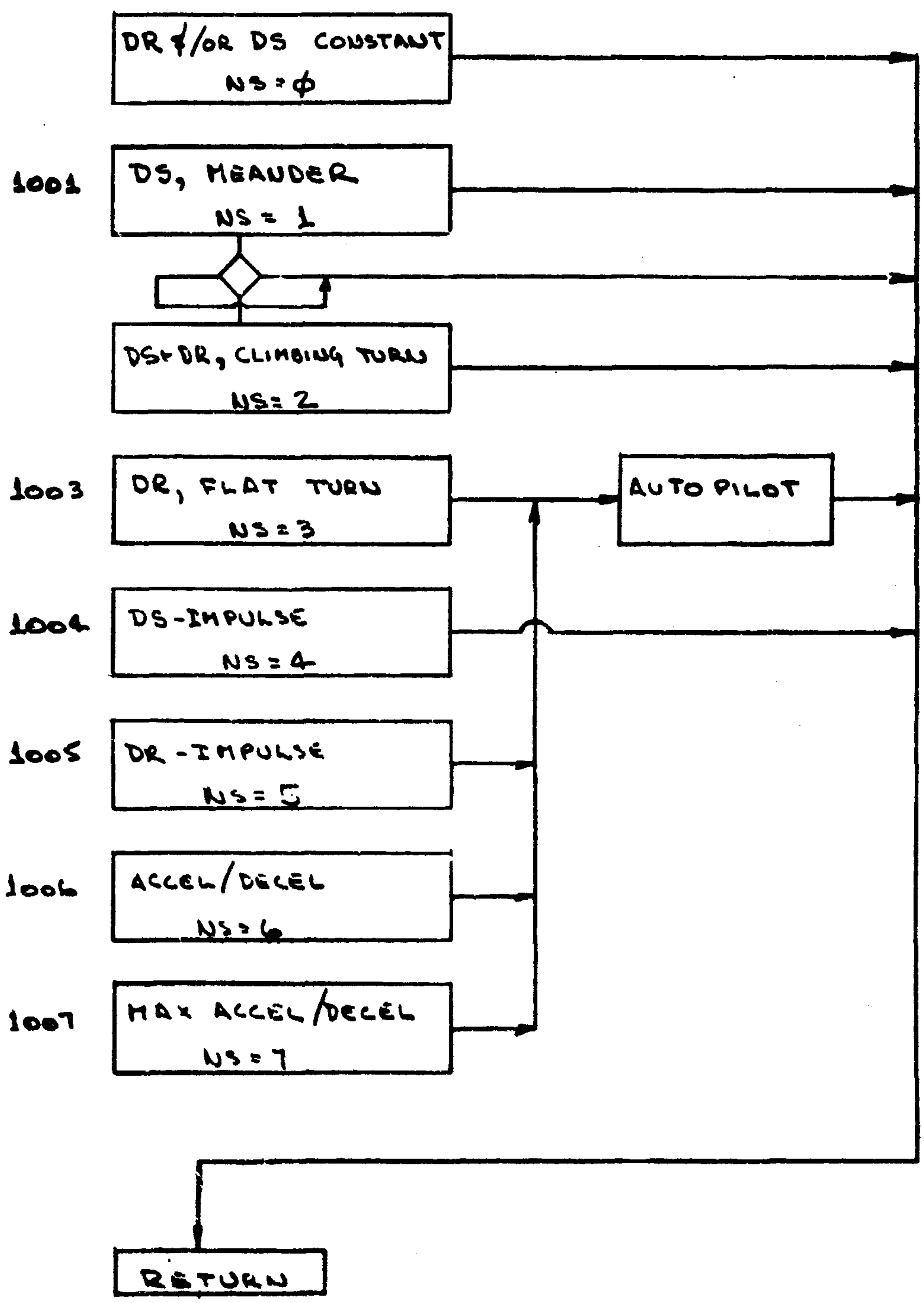


Figure 2. Subroutine CONTR, EB920, Block Diagram

TLIM defines the total time period of the runs in seconds.

Constant controls (NS = Ø) - The submarine will maintain the settings of command speed, (UC), elevator positions (DB and DS), or rudder position (DR) that were entered in the program as initial conditions. Thus, runs such as a steady turn without autopilot, climbing turn, acceleration without autopilot, etc., can be made.

Meander or Overshoot (NS = 1) - This control allows the submarine to maintain a period of level flight, followed by negative elevator movement at a specified rate to a minimum (largest negative value) elevator position as shown in figure 3. It will hold this angle until the submarine pitch angle reaches a particular value, SWMAX, at which time the elevators are reversed in position at a specified rate to a desired new position. The value of this position is determined by the type of run desired; for meanders, DELTMI = DS_o and for overshoot, DELTMA - DS_o equals DELTMI - DS_o in magnitude. The program allows variation of the various control parameters of this maneuver. These are:

- TIME - period of initial steady state performance, or constant input terms.
- R1 - negative rate of change of elevator position.
- DELTME - maximum negative swing of elevator (must be more negative than DS_o, elevator position at zero time).
- SWMAX - maximum dive angle, execute pitch angle allowed prior to turning elevators more positive. The time at which this occurs is referred to as T1.
- R2 - positive rate of change of elevator position.
- DELTMI - maximum position to which elevators are moved in the positive direction.

These input control parameters can be varied to achieve Meander, Overshoot, or other desired combinations of a dive to a maximum submarine pitch angle and subsequent elevator change.

Flat turn (NS = 3) - This is a turn accomplished by rudder movement in four steps to the maximum deflection, DRMAX. Each step is accomplished at a decreasing rate of change of rudder position as a function of the final maximum deflection, as shown in table 1. This schedule represents the delay in rudder position in the actual submarine.

TABLE 1. RUDDER POSITION SCHEDULE

Period	Rudder Position Rate of Change Radians/sec	Maximum Rudder Position Fraction of final position, DRMAX
1	.08726	.85
2	.01336	.93
3	.006	.97
4	.0001064	1.0

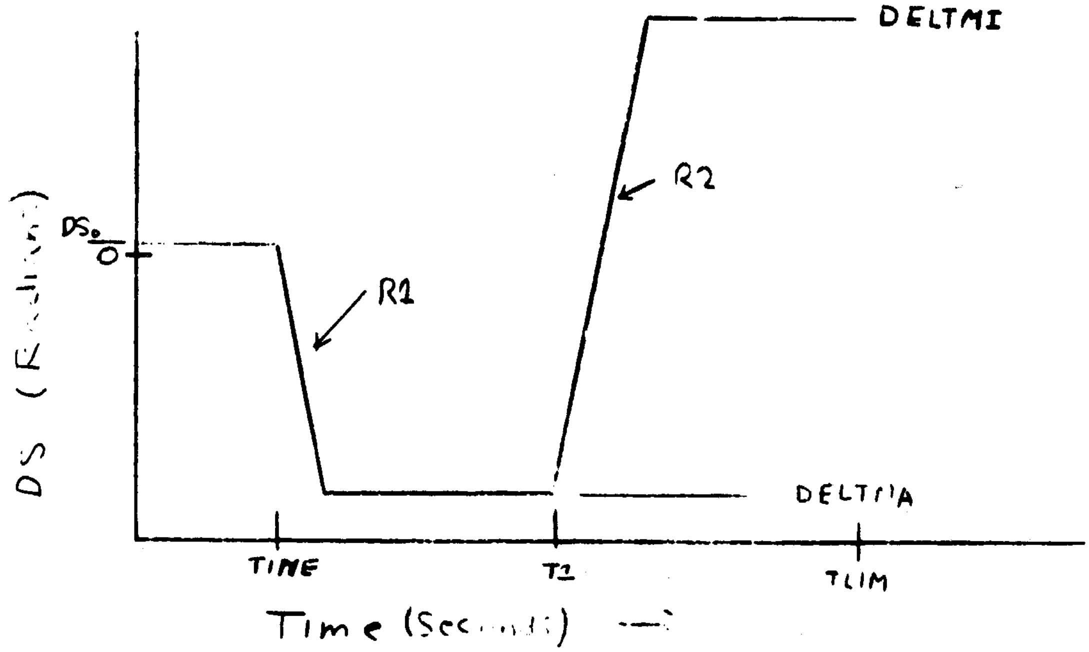


Figure 3. Graph of Overshoot Control

The program allows selection of the final rudder position, DRMAX. During this maneuver, the autopilot is actuated to maintain nearly level flight.

The autopilot is utilized in the control subroutine for certain maneuvers, such as flat turns, acceleration, etc. This control is automatic when used through selection of the desired control routine. The bow and stern planes are moved to attempt to maintain constant depth at any speed or turn condition. The elevators are positioned in accordance with equation (8).

$$DS = DB = .008\,(ZC - Z) + 3.5\,(\theta) + .012\,(U\sin\theta - W\cos\theta) + 2.0q \qquad (8)$$

Where:

- DB = sailplane angle, radians
- DS = sternplane angle, radians
- ZC = command depth, feet
- Z = depth, feet
- θ = pitch angle, radians
- u = forward body velocity, ft/sec
- w = normal body, velocity, ft/sec
- q = pitch rate, radians/sec

Position and rate damping in both depth and pitch are utilized. DS and DB are limited to 35 degrees by the programing.

Climbing turn (NS = 2) - This is a complex maneuver utilizing the available inputs of meander and flat turn above, with the autopilot inoperative. All the input data of each of these two other controls is required and the resultant submarine maneuver can obviously be rather unique.

DS - impulse (NS = 4) - This control allows an elevator impulse to be applied to the submarine for the first integration cycle in order to evaluate response frequency and damping factors. The initial conditions (UC, w, α, THETA, and Z) are entered as required. These are usually for level flight at some particular speed. The impulse value, elevator position (DS), is also entered. The final value of elevator position, DSF, is specified. This value is the elevator position for level flight at the selected speed.

A useful addition of the program for elevator-impulse control is the computer output of punched cards of the submarine pitch angle versus time. These are punched in 2E15.7 format (THETA, TIME) from T_0 to TLIM. The cards are useful in other programs associated with submarine response, described elsewhere in this report. Most associated programs are based on cards punched at two-second intervals. To achieve this the input value of integration interval, H, must be 0.25 seconds. This punch card data will be received for any of the integration methods selected through the option, INTSW. It is suggested that INTSW be set to '0' for impulse runs because the accuracy of other integration methods does not match the accuracy of the Runge Kutta method when calculating the violent initial maneuvers excited by an impulse run.

The value of DSF, after the impulse, is usually greater than zero. This is the elevator position required to maintain level flight at the steady-state speed of the submarine.

DR - impulse (NS = 5) - This control is identical to DS - impulse except that rudder position is used for this type run. In addition the final rudder position (DRF) is usually zero rather than a small finite value as for elevator impulse.

Acceleration/Deceleration Control (NS = 6) - This option takes the submarine at rest, at any depth; accelerates at command speed increments of five knots over each time increment, TIME (an input value) to 25 knots; and then decelerates in the same command speed increments. The autopilot control is activated in this run to maintain the submarine in nearly level flight.

Maximum Acceleration/Deceleration Control (NS = 7) - This control accelerates a submarine at rest, at input depth, Z, by applying a command speed of 25 knots. This command speed is held for TIME seconds (input value) which may be varied to assure steady state conditions. The submarine is then slowed down by reduction of command speed (UC) to zero for

the same length of time. The autopilot control is used for level flight.

d. EVAL - This subroutine calculates the right hand side of the equations of motion. The values are updated after each pass through the integration routine; Kutta. It follows the mathematical model given in Table 1.

e. PLOTROU - This subroutine transfers the run number, data names, and calculated points for storage on magnetic tape and subsequent plot of the variables on a California Computer plotter. This subroutine, in addition, calls out the following subroutines that are peculiar to Cal Camp software:

LINE PLOTS PLOT WHERE

AXIS NUMBER SYMBOL SCALE

These are all Cal Comp proprietary subroutines and thus cannot be supplied with this contract. If this program is run on a computer with the source decks supplied by GAC and Cal Comp plotting software is available minor program adjustments may be necessary to allow exact plotting as at GAC.

If this software is not available, three options are open:

(1) At GAC, the 360/40 computer will operate without the subroutines above, provided no plots are called for by leaving variable IPLOT = Ø on the first input data card. The linkage editor map at GAC shows "Unresolved entry message". However, the program still runs without plot. The program may run similarly on other computers.

(2) If this does not function on the computer used, dummy subroutines for these variables can be added to the programs. The program can then call them as at present and return. Leave IPLOT = Ø.

(3) Finally, the reference to all plotting subroutines could be removed from the program.

f. INVER2 - This is a common library routine to invert a matrix.

g. MATMPY - This subroutine takes the inverted matrix from INVER 2 and multiplies to calculate values of $\dot{u}$, $\dot{v}$, $\dot{w}$, $\dot{p}$, $\dot{q}$, and $\dot{r}$.

h. INTEG - This routine includes the three additional optional integration methods. These include:

(1) INTSW = 1, Second order Adams

$$Y_{n+1} = Y_n + \frac{h}{2}\left(3\dot{Y}_n - \dot{Y}_{n-1}\right)$$

(2) INTSW = 2, Second order non-classical

$$Y_{n+1} = Y_n + \frac{h}{4}\left(3\dot{Y}_n + \dot{Y}_{n-1}\right)$$

(3) INTSW = 3, Euler $Y_{n+1} = Y_n + h\dot{Y}_n$

where Y is the variable to be integrated

n is the number of times the variable was integrated

h is the integration time period.

i. EVAL 1 - This subroutine solves the right hand side of the equations of motion from data from each pass thru the integration routine, INTEG. It follows the mathematical model given in table 1.

3. INPUT DATA DECK

This section describes the data deck, defines the FORTRAN variables in terms of the equations of motion defined in the mathematical model, and shows how to perform the various kinds of simulated submarine operations allowed. The coefficients referred to here are those included in reference, "Standard Equations of Motion for Submarine Simulation". The coefficients are unique for each type submarine. These values are program inputs so that the program can be used for the study of different types of submarines. The input deck should follow table 2 exactly. Figure 4 shows a typical input data form for EB920.

TABLE 2. INPUT DATA DECK, PROGRAM EB920

Card	Column(s)	Format	Description
Control Flags 1	1-5	I5	NGS. The number of good integration steps required before step size is increased. Blank if variable step size is not used. (Note right adjust all integer values)
1	6-10	I5	NPNT. Data will print out at T_o and each NPNTth integration step (Each 2 sec if NPNT = 8 and H = .25 seconds)
1	11-15	I5	IPLOT. To exercise plot option, IPLOT 1. Leave blank (zero) for no plots.

TABLE 2. INPUT DATA DECK, PROGRAM EB920 (cont.)

Card	Column(s)	Format	Description
1	16-20	I5	IRUN. Identification number for individual runs. If IRUN = 0 a normal exit is made
1	21-25	I5	NPLT. Data will plot at T_o and each NPLT$^{\underline{th}}$ integration step.
1	26-30	I5	IOPT. This option will allow changing any one or more of the input values for a succeeding run without putting in all the other values. IOPT = 1 to exercize; blank for no selection. Another run may still follow, for IOPT = Ø, but all input data cards must be read again. More information on this variable is included at card 33.
1	31-35	I5	ICYC. Number of integration cycles per H, time increment. ICYC = 4 for Kutta integration (INTSW = Ø), = 1 for all other integration methods.
1	36-40	I5	NS. This variable selects the type of submarine control in CONTR subroutine: NS = 0 Fixed controls per initial conditions NS = 1 Overshoot, meander, etc. NS = 2 Special climbing turn NS = 3 Flat turn (with autopilot) NS = 4 Elevator impulse NS = 5 Rudder impulse (with autopilot) NS = 6 Acceleration/deceleration (with autopilot) NS = 7 Maximum acceleration/deceleration (with autopilot)
1	41-45	I5	INTSW. This variable selects its type of integration to be used. INTSW = 0 Runge-Kutta

TABLE 2. INPUT DATA DECK, PROGRAM EB920 (cont.)

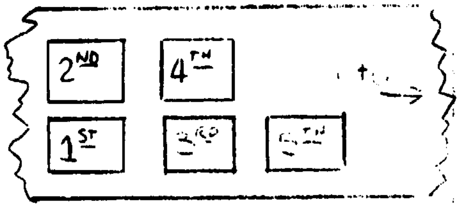
Card	Column(s)	Format	Description
			INTSW = 1 2nd order Adams INTSW = 2 2nd order Non-classical INTSW = 3 Euler
Plot Control 2	 1-75	 15I5	Card must be here (blank), even if no plots are required. ILOC (I), I=2,16. Value of I defines which variable will be plotted against time. Sequence defines order of plot on plotting paper as:  Order of plotting Plot Variable: ILOC (2) - U, component of velocity in the x-direction, feet/second. ILOC (3) - V, component of velocity in the y-direction, feet/second. ILOC (4) - W, component of velocity z-direction, feet/second. ILOC (5) - P, angular velocity component about the x-axis, radians/second ILOC (6) - Q, angular velocity component about the y-axis, radians/second ILOC (7) - R, angular velocity component about the z-axis radians/second ILOC (8) - THETA (θ), pitch angle, radians ILOC (9) - PSI (ψ), yaw angle, radians ILOC (10) - PHI (ϕ), roll angle, radians ILOC (11) - X, coordinate

TABLE 2. INPUT DATA DECK, PROGRAM EB920 (cont.)

Card	Column(s)	Format	Description
			point of sub position, feet ILOC (12) - Y, coordinate point of sub position, feet ILOC (13) - Z, coordinate point of sub position, (depth), feet ILOC (14) - DR, rudder position, radians ILOC (15) - DS, stern elevator position, radians ILOC (16) - DB, bow elevator position, radians
Timing			
3	1-10	F10.5	TO, starting time, seconds (usually zero)
3	11-20	F10.5	HO, initial integration time increment (step) size. If DH is zero, this step size will be used for the entire run.
3	21-30	F10.5	DH, factor by which the step size is modified. If DH is 2, the step size is doubled or halved as required.
3	31-40	F10.5	HMAX, maximum step size allowed
3	41-50	F10.5	HMIN, minimum step size allowed
3	51-60	F10.5	FCT, a factor which causes the error estimate corresponding to the next larger step size to be overestimated. This reduces time-consuming premature increases in step size. Typical value is 1.2.
3	61-70	F10.5	TLIM, time of run, seconds
4	1-80	8F10.5	TL (I), U, V, W, P, Q, R, Θ, Ψ. Tolerance array for variable integration step size. Allowable deviation of each of these parameters at any point calculated. Card must be present even if variable step size is not used

TABLE 2. INPUT DATA DECK, PROGRAM EB920 (cont.)

Card	Column(s)	Format	Description
5	1-40	4F10.5	TL (I), Ø, X, Y, Z. Allowable deviation of these parameters.
Initial Conditions Array			
6	1-10	F10.5	Y(1), U, velocity in x-direction, feet/second
6	11-20	F10.5	Y(2), V, velocity in y-direction, feet/second
6	21-30	F10.5	Y(3), W, velocity in z-direction, feet/second
6	31-40	F10.5	Y(4), P, velocity about x-axis, radians/second
6	41-50	F10.5	Y(5), Q, velocity about y-axis, radians/second
6	51-60	F10.5	Y(6), R, velocity about z-axis, radians/second
6	61-70	F10.5	Y(7), THETA, pitch angle,radians
6	71-80	F10.5	Y(8), PSI, yaw angle, radians
7	1-10	F10.5	Y(9), PHI, roll angle, radians
7	11-20	F10.5	Y(10), X, coordinate point of sub position, feet
7	21-30	F10.5	Y(11), Y, coordinate point of sub position, feet
7	31-40	F10.5	Y(12), Z, coordinate point of sub position (depth), feet
Coefficient Cards			
8	1-80	8F10.5	XQQ, XPR, XRP, XUD, XVR, XWQ, XUU, XVV
9	1-80	8F10.5	XWW, XDRDR, XDSDS, XDBDB, XVVE, XWWE, XDRDRE, XDSDSE
10	1-80	8F10.5	YRD, YPD, YPAP. YPQ, YQR, YVD, YVQ, YWP
11	1-80	8F10.5	YWR, YR, YP, YARDR, YVAR, YSTR,

TABLE 2. INPUT DATA DECK, PROGRAM EB920 (cont.)

Card	Column(s)	Format	Description
			YV, YVAV
12	1-80	8F10.5	YVW, YDR, YRE, YVE, YVAVE, YDRE
13	1-80	8F10.5	ZQD, ZPP, ZRR, ZRP, ZWD, ZVR, ZVP, ZQ
14	1-80	8F10.5	ZAQDS, ZWAQ, ZSTR, ZW, ZWAW, ZAW, ZWW, ZVV
15	1-80	8F10.5	ZDS, ZDB, ZQE, ZWE, ZWAWE, ZDSE
16	1-80	8F10.5	AKPD, AKRD, AKQR, AKPQ, AKPAP, AKP, AKR, AKVD
17	1-80	8F10.5	AKVQ, AKWP, AKWR, AKSTR, AKV, AKVAV, AKVW, AKDR
18	1-80	8F10.5	AKSTRE
19	1-80	8F10.5	AMQD, AMPP, AMRR, AMRP, AMQAQ, AMWD, AMVR, AMVP
20	1-80	8F10.5	AMQ, AMAQDS, AMAWQ, AMSTR, AMW, AMWAW, AMAW, AMWW
21	1-80	8F10.5	AMVV, AMDS, AMDB, AMQE, AMWE, AMWAWE, AMDSE
22	1-80	8F10.5	ANRD, ANPD, ANPQ, ANQR, ANRAR, ANVD, ANWR, ANWP
23	1-80	8F10.5	ANVQ, ANP, ANR, ANARDR, ANAVR, ANSTR, ANV, ANVAV
24	1-80	8F10.5	ANVW, ANDR, ANRE, ANVE, ANVAVE, ANDRE
Submarine Constants			
25	1-10	F10.5	IX, moment of inertia about the x-axis, slug-ft^2
25	11-20	F10.5	IY, moment about y-axis, slug-ft^2
25	21-30	F10.5	IZ, moment about z-axis, slug-ft^2
25	31-40	F10.5	IXY, product of inertia about xy-axis, (slug-ft^2)^2

TABLE 2. INPUT DATA DECK, PROGRAM EB920 (cont.)

Card	Column(s)	Format	Description
25	41-50	F10.5	IXZ, product of inertia about xz-axis, $(\text{slug-ft}^2)^2$
25	51-60	F10.5	IYZ, product of inertia about yz-axis, $(\text{slug-ft}^2)^2$
26	1-10	F10.5	CW, weight of submarine, including water in free-flooding space, pounds
26	11-20	F10.5	CB, buoyancy, pounds
26	21-30	F10.5	UC, initial command speed, feet/second
26	31-40	F10.5	XB, x-component of center of buoyancy, feet
26	41-50	F10.5	YB, y-component of center of buoyancy, feet
26	51-60	F10.5	ZB, z-component of center of buoyancy, feet
27	1-10	F10.5	DR, initial value of rudder position, radians
27	11-20	F10.5	DS, initial value of stern elevator position, radians
27	21-30	F10.5	DB, initial value of bow elevator position, radians
27	31-40	F10.5	RHO, density of sea water, slugs/feet^3
27	41-50	F10.5	AL, submarine length, feet
27	51-60	F10.5	AM, submarine mass, including water in free flooding space, slugs
28	1-10	F10.5	DRMAX, maximum rudder position (movement), radians
28	11-20	F10.5	ETAHI(η-high), upper reference value of UC/UMAG, dimensionless
28	21-30	F10.5	ETALO(η-low), lower reference value of UC/UMAG, dimensionless

TABLE 2. INPUT DATA DECK, PROGRAM EB920 (cont.)

Card	Column(s)	Format	Description
28	31-40	F10.5.	A11, value of a_i for ETA ETAHI
28	41-50	F10.5	A12, value of b_i for ETA ETAHI
28	51-60	F10.5	A13, value of c_i for ETA ETAHI
29	1-10	F10.5	A21, value of a_i for ETALO ETA ETAHI
29	11-20	F10.5	A22, value of b_i for ETALO ETA ETAHI
29	21-30	F10.5	A23, value of c_i for ETALO ETA ETAHI
29	31-40	F10.5	A31, value of a_i for ETA ETALO
29	41-50	F10.5	A32, value of b_i for ETA ETALO
29	51-60	F10.5	A33, value of c_i for ETA ETALO
30	1-10	F10.5	XG, x-component of center of gravity, feet
30	11-20	F10.5	YG, y-component of center of gravity, feet
30	21-30	F10.5	ZG, z-component of center of gravity, feet
Surface Control Schedule			
31	1-10	F10.5	TIME, various periods of time depending on NS in control routine, seconds NS — TIME 0,2,3,4,5 — Not used 1 — Initial steady state period (See fig.4) 6 — Period of constant application of each command speed 7 — Period of constant application of each

TABLE 2. INPUT DATA DECK, PROGRAM EB920 (cont.)

Card	Column(s)	Format	Description
			command speed
31	11-20	F10.5	R1, negative rate of change of elevator position, radians/second during meander runs (NS = 1), radians/second (See fig.4)
31	21-30	F10.5	DELTMA, maximum negative swing of elevator (must be more negative than DS, initial condition) during meander runs (NS = 1), radians (See fig. 4)
31	31-40	F10.5	SWMAX (Θ'), execute pitch angle, maximum submarine dive angle during meander-type run (NS = 1) before turning elevators to a more positive value, radians.
31	41-50	F10.5	R1, positive rate of change of elevator position after submarine reaches Θ', radians/second (See figure 4)
31	51-60	F10.5	DELTMI, maximum position to which elevators are moved during 'R2' change, radians (See fig.4)
31	61-70	F10.5	DSF, final value of elevator position during DS-impulse run (NS = 2), radians
31	71-80	F10.5	DRF, final value of rudder position during DR-impulse run (NS = 5), radians
Additional Run Controls			If IOPT was "O" on card 1 of the last data deck, card 32 can be either blank for a normal exit or card 1 of a new data deck for additional runs. If IOPT = F on card 1 of the last data deck, see below.
32	1-5	I5	IRUN, run numbers. IRUN = 0 (blank card) a new data deck is read. A blank card at the start of a data deck results in a normal exit so two blank cards at the end

TABLE 2. INPUT DATA DECK, PROGRAM EB920 (cont.)

Card	Column(s)	Format	Description
			will always end the run. IRUN = Integer. Use as new run number and continue reading cards below.
33	1-5	I5	NDEX, common location for parameter to be changed. Tables 3 and 4 show the variable names versus the common locations, NDEX.
33	11-20	F10.5	VALUE, new value of parameter changed in columns 1-5 of this card.
33 + n	1-20	I5, 5X F10.5	Repeat card 33 as desired.
33 + n + 1	-	Blank	Start new run after all changes have been made.

IBM GENERAL PURPOSE CARD PUNCHING FORM

PUNCHING INSTRUCTIONS

JOB EB920		WRITTEN AS:
BY	DATE	PUNCH AS

NOTES Data Deck Format, EB920. RIGHT ADJUST ALL INTEGERS.

Card	1-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80
1	NGS NPNT	IPLOT IKUN	NPLT IOPT	ICYC NS	INTSW			
2	ILOC ILOC	ILOC ILOC	ILOC ILOC	ILOC ILOC	ILOC ILOC	ILOC ILOC	ILOC ILOC	ILOC
3	T9	HO	DH	HMAX	HMIN	FLT	TLIM	
4	U	V	W	P	Q	R	THETA	PSI
5	PHI	X	Y	Z				
6	U	V	W	P	Q	R	THETA	PSI
7	PHI	X	Y	Z				
8	X2Q	XKR	XRP	XUD	XVR	XWQ	XUU	XVV
9	XWW	XURDR	XDSDS	XDBDB	XVVE	XWWE	XDRDRE	XDSDSE
10	YRD	Y?D	YPAP	YPQ	YQR	YVD	YVQ	YWP
11	YWR	YR	YP	YRPDR	YVAR	YSTR	YV	YVAV
12	YVW	YDR	YKE	YVE	YVAVE	YDRE		
13	ZQD	ZFP	ZRR	ZRP	ZWD	ZVR	ZVP	ZQ
14	ZAQDS	ZWAQ	ZSTR	ZW	ZWAW	ZAW	ZWW	ZVV
15	ZDS	ZDB	ZQE	ZWE	ZWAWE	ZDSE		
16	AKFD	AKRD	AKQR	AKPQ	AKFAP	AKP	AKR	AKVD
17	AKVQ	AKWF	AKWR	AKSTR	AKV	AKVAV	AKVW	AKDR
18	AKSTRE							
19	AMQD	AMPF	AMKR	AMKP	AMQRQ	AMWC	AMVR	AMVP
20	AMQ	AMAQDS	AMAWQ	AMSTR	AMW	AMWAW	AMAW	AMWW

Printed in U.S.A. X20-8030-03 UM/025 PAGE ___ OF ___

Figure 4. Input Data Form, Program EB920

IBM GENERAL PURPOSE CARD PUNCHING FORM

JOB		
BY	DATE	

PUNCHING INSTRUCTIONS

WRITTEN AS:												
PUNCH AS:												

NOTES

FIELD IDENTIFICATION

	1-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80
21	AMVV	AMDS	AMDB	AMQE	AMWE	AMWAWE	AMDSE	
22	ANRD	ANPD	ANPQ	ANQR	ANRAR	ANVD	ANWR	ANWP
23	ANVQ	ANP	ANR	ANARDR	ANAVR	ANSTR	ANV	ANVAV
24	ANVW	ANDR	ANRE	ANVE	ANVAVE	ANDRE		
25	IX	IY	IZ	IXY	IXZ	IYZ		
26	CW	CB	UC	XB	YB	ZB		
27	DR	DS	DB	RHO	AL	AM		
28	DRMAX	ETAHI	ETALO	A11	A12	A13		
29	A21	A22	A23	A31	A32	A33		
30	XG	YG	ZG					
31	TIME	R1	DELTMA	SWMAX	R2	DELTMI	DSF	DRF
32	IRUN							
33	INDEX	VALUE						
14								
15								
16								
17								
18								
19								
20								

Printed in U.S.A. X20-8030-03 UM/025 PAGE ___ OF ___

Figure 4. Input Data Form, Program EB920 (cont.)

TABLE 3. EB920, COMMON LAYOUT INDEX

Location	Term	Location	Term	Location	Term	Location	Term
1	H	36	XWQ	71	ZRR	106	AKDR
2	HMAX	37	XUU	72	ZRP	107	AKSTRE
3	HMIN	38	XVV	73	ZWD	108	AMQD
4	DH	39	XWW	74	ZVR	109	AMPP
5	FCT	40	XDRDR	75	ZVP	110	AMRR
6	TL(1)	41	XDSDS	76	ZQ	111	AMRP
7	TL(2)	42	XDBDB	77	ZAQDS	112	AMQAQ
8	TL(3)	43	XVVE	78	ZWAQ	113	AMWD
9	TL(4)	44	XWWE	79	ZSTR	114	AMVR
10	TL(5)	45	XDRDRE	80	ZW	115	AMVP
11	TL(6)	46	XDSDSE	81	ZWAW	116	AMQ
12	TL(7)	47	YRD	82	ZAW	117	AMAQDS
13	TL(8)	48	YPD	83	ZWW	118	AMAWQ
14	TL(9)	49	YPAP	84	ZVV	119	AMSTR
15	TL(1Ø)	50	YPQ	85	ZDS	120	AMW
16	TL(11)	51	YQR	86	ZDB	121	AMWAW
17	TL(12)	52	YVD	87	ZQE	122	AMAW
18	NGS	53	YVQ	88	ZWE	123	AMWW
19	N	54	YWP	89	ZWAWE	124	AMVV
20	IS1	55	YWR	90	ZDSE	125	AMDS
21	NPNT	56	YR	91	AKPD	126	AMDB
22	TLIM	57	YP	92	AKRD	127	AMQE
23	RHØL2	58	YARDR	93	AKQR	128	AMWE
24	RHØL3	59	YVAR	94	AKPQ	129	AMWAWE
25	RHØL4	60	YSTR	95	AKPAP	130	AMDSE
26	RHØL5	61	YV	96	AKP	131	ANRD
27	WMB	62	YVAV	97	AKR	132	ANPD
28	ETA	63	YVW	98	AKVD	133	ANPQ
29	ETAML	64	YDR	99	AKVQ	134	ANQR
30	ISW2	65	YRE	100	AKWP	135	ANRAR
31	XQQ	66	YVE	101	AKWR	136	ANVD
32	XRR	67	YVAVE	102	AKSTR	137	ANWR
33	XRP	68	YDRE	103	AKV	138	ANWP
34	XUD	69	ZQD	104	AKVAV	139	ANVQ
35	XVR	70	ZPP	105	AKVW	140	ANP

TABLE 3. EB920, COMMON LAYOUT INDEX (cont.)

Location	Term	Location	Term	Location	Term	Location	Term
141	ANR	176	A13	211	Q		
142	ANARDR	177	A21	212	R		
143	ANAVR	178	A22	213	Θ		
144	ANSTR	179	A23	214			
145	ANV	180	A31	215			
146	ANVAV	181	A32	216	X		
147	ANVW	182	A33	217	Y		
148	ANDR	183	XG	218	Z		
149	ANRE	184	YG	219	TIME		
150	ANVE	185	ZG	220	R1		
151	ANVAVE	186	ILΘC(1)	221	DELTMA		
152	ANDRE	187	ILΘC(2)	222	SWMAX		
153	IX	188	ILΘC(3)	223	R2		
154	IY	189	ILΘC(4)	224	DELTMI		
155	IZ	190	ILΘC(5)	225	DSF		
156	IXY	191	ILΘC(6)	226	DRF		
157	IXZ	192	ILΘC(7)	227	ICYC		
158	IYZ	193	ILΘC(8)	228	NS		
159	CW	194	ILΘC(9)	229	INTSW		
160	CB	195	ILΘC(10)				
161	UC	196	ILΘC(11)				
162	XB	197	ILΘC(12)				
163	YB	198	ILΘC(13)				
164	ZB	199	ILΘC(14)				
165	DR	200	ILΘC(15)				
166	DS	201	ILΘC(16)				
167	DB	202					
168	RHΘ	203					
169	AL	204					
170	AM	205					
171	DRMAX	206					
172	ETAHI	207					
173	ETALΘ	208					
174	A11	209					
175	A12	210					

TABLE 4. EB920, COMMON LAYOUT, ALPHABETICAL

Location	Term	Location	Term	Location	Term	Location	Term
174	A11	195	ILOC(10)	170	M	131	NRD
175	A12	196	ILOC(11)	117	MAQDS	149	NRE
176	A13	197	ILOC(12)	122	MAW	228	NS
177	A21	198	ILOC(13)	118	MAWQ	144	NSTR
178	A22	199	ILOC(14)	126	MDB	145	NV
179	A23	200	ILOC(15)	125	MDS	146	NVAN
180	A31	201	ILOC(16)	130	MDSE	151	NVAVE
181	A32	229	INTSW	109	MPP	136	NVD
182	A33	204	IOPEN	116	MQ	150	NVE
160	CB	206	IOPT	112	MQAQ	139	NVQ
159	CW	202	IPLOT	108	MQD	147	NVW
167	DB	203	IRUN	127	MQE	138	NWP
221	DELTMA	20	IS1	111	MRP	137	NWR
224	DELTMI	30	ISW2	110	MRR	210	P
4	DH	153	IX	119	MSTR	215	PHI
165	DR	156	IXY	115	MVP	214	PSI
226	DRF	157	IXZ	114	MVR	211	Q
171	DRMAX	154	IY	124	MVV	212	R
166	DS	158	IYZ	120	MW	168	RHO
225	DSF	155	IZ	121	MWAW	23	RHOL2
28	ETA	106	KDR	129	MWAWE	24	RHOL3
172	ETAHI	96	KP	113	MWD	25	RHOL4
173	ETALO	95	KPAP	128	MWE	26	RHOL5
29	ETAMI	91	KPD	123	MWW	220	R1
5	FCT	94	KPQ	19	N	223	R2
1	H	93	KQR	143	NAVR	222	SWMAX
2	HMAX	97	KR	142	NARDR	213	THETA
3	HMIN	92	KRD	148	NDR	219	TIME
227	ICYC	102	KSTR	152	NDRE	6	TL(1)
186	ILOC(1)	107	KSTRE	18	NGS	7	TL(2)
187	ILOC(2)	103	KV	140	NP	8	TL(3)
188	ILOC(3)	104	KVAV	132	NPD	9	TL(4)
189	ILOC(4)	98	KVD	205	NPLT	10	TL(5)
190	ILOC(5)	99	KVQ	21	NPNT	11	TL(6)
191	ILOC(6)	105	KVW	133	NPQ	12	TL(7)
192	ILOC(7)	100	KWP	134	NQR	13	TL(8)
193	ILOC(8)	101	KWR	141	NR	14	TL(9)
194	ILOC(9)	169	L	135	NRAR	15	TL(10)

TABLE 4. EB920, COMMON LAYOUT, ALPHABETICAL (cont.)

Location	Term	Location	Term	Location	Term	Location	Term
16	TL(11)	56	YR	83	ZWW		
17	TL(12)	47	YRD				
22	TLIM	65	YRE				
207	U	60	YSTAR				
161	UC	61	YV				
208	V	59	YVAR				
209	W	62	YVAV				
27	WMB	67	YVAVE				
216	X	52	YVD				
162	XB	66	YVE				
42	XDBDB	53	YVQ				
40	XDRDR	63	YVW				
45	XDRDRE	54	YWP				
41	XDSDS	55	YWR				
46	XDSDSE	218	Z				
183	XG	77	ZAQDS				
31	XQQ	82	ZAW				
33	XRP	164	ZB				
32	XRR	86	ZDB				
34	XUD	85	ZDS				
37	XUU	90	ZDSE				
35	XVR	185	ZG				
38	XVV	70	ZPP				
43	XVVE	76	ZQ				
36	XWQ	69	ZQD				
39	XWW	87	ZQE				
44	XWWE	72	ZRP				
217	Y	71	ZRR				
58	YARDR	79	ZSTR				
163	YB	75	ZVP				
64	YDR	74	ZVR				
68	YDRE	84	ZVV				
184	YG	80	ZW				
57	YP	78	ZWAQ				
49	YPAP	81	ZWAW				
48	YPD	89	ZWAWE				
50	YPQ	73	ZWD				
51	YQR	88	ZWE				

4. OUTPUT DATA

a. Printout - This program will print out titled numbered pages with the variable as shown in table 5. All input values are printed out at the start of the run as shown in figure 5.

TABLE 5. OUTPUT VARIABLES, PROGRAM EB920

Variable	Format	Units	Description
U	E13.6	feet/second	Velocity component in the x-direction
V	E13.6	feet/second	Velocity component in the y-direction
W	E13.6	feet/second	Velocity component in the z-direction
P	E13.6	radians/second	Velocity about x-axis
Q	E13.6	radians/second	Velocity about y-axis
R	E13.6	radians/second	Velocity about z-axis
THETA	E13.6	radians	Pitch angle
PSI	E13.6	radians	Yaw angle
PHI	E13.6	radians	Yaw angle
X	E13.6	feet	Coordinate point of sub position
Y	E13.6	feet	Coordinate point of sub position
Z	E13.6	feet	Coordinate point of sub position
T	E13.6	seconds	Time during run
H	E13.6	seconds	Integrating time period

All output variables are printed as shown in sample data sheet, figure 6. This data is a function of the time of the run, T. The frequency of printout is a function of NPNT, printed each NPNth integration cycle. If H, integration time period is 0.25 second and NPNT = 8:

a. INTSW = Ø (Runge-Kutta integration)

$$T'' = \frac{NPNT \times H}{ICYC} = \frac{8 \times 0.25}{4} = 0.5 \text{ seconds}$$

Printout would be each half second

EB920 SUBMARINE SIMULATION PAGE 1

XQQ	-0.10000E-03	YRD	0.10000E-03	ZQD	-0.10000E-03	KPD	-0.20000E-04	MQD	-0.70000E-03	NRD	-0.75000E-03
XRR	-0.10000E-03	YPD	-0.75000E-04	ZPP	0.0	KRD	-0.18000E-04	MPP	0.0	NPD	-0.20000E-04
XRP	0.75000E-04	YPAP	0.0	ZRR	-0.40000E-02	KQR	-0.33000E-04	MRR	-0.20000E-02	NPQ	-0.70000E-03
XUD	-0.40000E-03	YPQ	0.13000E-03	ZRP	-0.10000E-02	KPQ	0.0	MRP	0.70000E-03	NQR	0.0
XVR	0.15000E-01	YQR	0.0	ZWD	-0.14000E-01	KPAP	-0.20000E-05	MQAQ	0.0	NRAR	0.0
XWQ	-0.14000E-01	YVD	-0.15000E-01	ZVR	0.80000E-02	KP	-0.70000E-04	MWD	-0.13000E-03	NVD	0.20000E-05
XUU	0.0	YVQ	0.0	ZVP	-0.15000E-01	KR	0.0	MVR	-0.40000E-02	NWP	0.0
XVV	0.15000E-01	YWP	0.15000E-01	ZQ	-0.70000E-02	KVD	-0.86000E-04	MVP	0.10000E-02	NWP	-0.13000E-03
XWW	0.50000E-02	YWR	0.0	ZAQDS	0.30000E-02	KVQ	0.0	MQ	-0.40000E-02	NVQ	0.0
XDRDR	-0.50000E-02	YR	0.40000E-02	ZWAQ	-0.10000E-01	KWP	0.86000E-04	MAQDS	-0.50000E-03	NP	-0.20000E-03
XDSDS	-0.30000E-02	YP	-0.70000E-03	ZSTR	0.22000E-03	KWR	0.0	MAWQ	-0.30000E-02	NR	-0.40000E-02
XDBDB	-0.50000E-02	YARDR	-0.30000E-02	ZW	-0.17000E-01	KSTR	0.0	MSTR	0.75000E-04	NARDR	-0.55000E-03
XVVF	0.0	YVAR	-0.10000E-01	ZWAW	-0.55000E-01	KV	-0.20000E-03	MW	0.75000E-02	NAVR	-0.33000E-02
XWWF	0.0	YSTR	0.0	ZAW	0.0	KVAV	-0.23000E-02	MWAW	-0.14000E-01	NSTR	0.0
XDRDRF	0.0	YV	-0.20000E-01	ZWW	0.0	KVW	0.0	MAW	0.0	NV	-0.10000E-01
XDSDSF	0.0	YVAV	-0.75000E-01	ZVV	0.90000E-01	KDR	0.0	MWW	0.0	NVAV	0.12000E-01
		YVW	-0.10000E 00	ZDS	-0.85000E-02	KSTRF	0.0	MVV	0.14000E-01	NVW	0.14000E-01
		YDR	0.90000E-02	ZDB	-0.50000E-02			MDS	-0.40000E-02	NDR	-0.43000E-02
		YRF	0.0	ZQF	0.0			MDB	0.10000E-02	NRF	0.0
		YVF	0.0	ZWF	0.0			MQF	0.0	NVF	0.0
		YVAVF	0.0	ZWAWF	0.0			MWF	0.0	NVAVF	0.0
		YDRF	0.0	ZDSF	0.0			MWAWF	0.0	NDRF	0.0
								MDSF	0.0		
IX	0.36200E 08	IY	0.13000E 10	IZ	0.13000E 10	IXY	0.0	IXZ	0.0	IYZ	0.0
W	0.10000E 08	B	0.10000E 08	UC	0.84450E 01	XB	0.0	YB	0.0	ZB	-0.16000E 01
DP	0.0	DS	0.15254E-01	DB	0.0	RHO	0.20000E 01	L	0.27500E 03	M	0.30000E 05
A11	-0.15000E-02	A21	-0.20500E-02	A31	-0.36400E-02	DRMAX	0.0	ETAHI	0.50000E 00	ETALO	-0.50000E 00
A12	-0.13000E-02	A22	0.12000E-02	A32	-0.39200E-02	XG	0.0	YG	0.0	ZG	0.0
A13	-0.24000E-02	A23	0.0	A33	-0.30850E-02	H	0.25000E 00	INTSW	1		
TIME	0.50000E 01	RI	-0.87270E-01	DELTMA	-0.34906E 00	SWMAX	0.13963E 00	R2	0.87270E-01	DELTMI	0.37957E 00
DSF	0.0	DRF	0.0	ICYC	1	NS	1				

RUN NO 1

Figure 5. Input Variable Format, EB920

EB920 SUBMARINE SIMULATION PAGE 2

U / PSI	V / PHI	W / X	P / Y	Q / Z	R / T	THETA / H
0.844500E 01	0.0	0.435340E-01	0.0	0.0	0.0	0.515500E-02
0.0	0.0	0.0	0.0	0.800000E 03	0.0	
0.804808E 01	0.0	0.434678E-01	0.0	-0.106898E-04	0.0	0.513447E-02
0.0	0.0	0.340176E 02	0.0	0.800002E 03	0.400000E 01	
0.767321E 01	0.0	0.737976E-01	0.0	0.867065E-03	0.0	0.603682E-02
0.0	0.0	0.654605E 02	0.0	0.800040E 03	0.800000E 01	
0.729265E 01	0.0	0.166942E 00	0.0	0.308199E-02	0.0	0.141125E-01
0.0	0.0	0.953937E 02	0.0	0.800238E 03	0.120000E 02	
0.692080E 01	0.0	0.258159E 00	0.0	0.445900E-02	0.0	0.294925E-01
0.0	0.0	0.123829E 03	0.0	0.800484E 03	0.160000E 02	
0.655773E 01	0.0	0.339740E 00	0.0	0.501513E-02	0.0	0.486994E-01
0.0	0.0	0.150810E 03	0.0	0.800634E 03	0.200000E 02	
0.620413E 01	0.0	0.408517E 00	0.0	0.487912E-02	0.0	0.686986E-01
0.0	0.0	0.176374E 03	0.0	0.800635E 03	0.240000E 02	
0.586158E 01	0.0	0.463152E 00	0.0	0.419937E-02	0.0	0.870135E-01
0.0	0.0	0.200564E 03	0.0	0.800496E 03	0.280000E 02	
0.553182E 01	0.0	0.503725E 00	0.0	0.313327E-02	0.0	0.101783E 00
0.0	0.0	0.223478E 03	0.0	0.800270E 03	0.320000E 02	
0.521614E 01	0.0	0.531305E 00	0.0	0.183672E-02	0.0	0.111775E 00
0.0	0.0	0.245017E 03	0.0	0.800032E 03	0.360000E 02	
0.491503E 01	0.0	0.547555E 00	0.0	0.455154E-03	0.0	0.116364E 00
0.0	0.0	0.265383E 03	0.0	0.799863E 03	0.400000E 02	
0.462818E 01	0.0	0.553561E 00	0.0	-0.878987E-03	0.0	0.115477E 00
0.0	0.0	0.284594E 03	0.0	0.799836E 03	0.440000E 02	
0.435445E 01	0.0	0.548960E 00	0.0	-0.205085E-02	0.0	0.109549E 00
0.0	0.0	0.302588E 03	0.0	0.800004E 03	0.480000E 02	
0.409214E 01	0.0	0.536467E 00	0.0	-0.299057E-02	0.0	0.993775E-01
0.0	0.0	0.319710E 03	0.0	0.800396E 03	0.520000E 02	
0.383933E 01	0.0	0.518833E 00	0.0	-0.365497E-02	0.0	0.859876E-01
0.0	0.0	0.335696E 03	0.0	0.80107[illegible]E 03	0.560000E 02	

Figure 6. Output Variable Format, EB920

b. INTSW = 1 (Adams integration)

$$T'' = \frac{NPNT \times H}{LCYC} = \frac{8 \times 0.25}{1} = 2.0 \text{ seconds}$$

Printout would be every two seconds.

c. Graphical - This program can optionally plot any of the printed values noted in Section 4.a on a CALCOMP plotter. In addition, the control parameters, DS, DB, and DR (elevator and rudder positions) can be plotted. These variables are each plotted (as requested) as a function of time as shown in figure 7.

5. LISTINGS AND FLOW CHART

The listings and flow chart for program EB920 are given in appendix A.

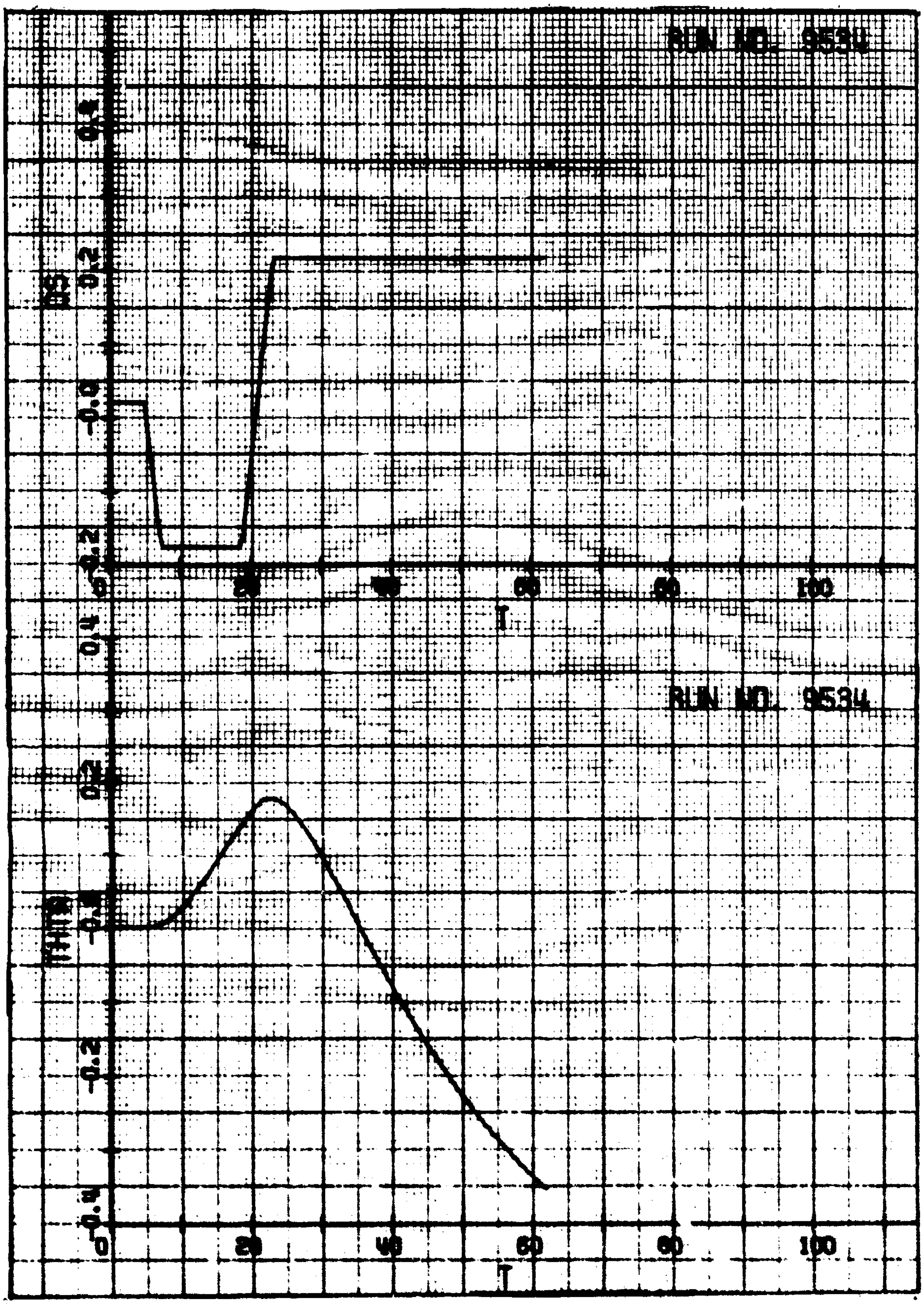


Figure 7. Graphical Output EB920

B. PROGRAM ZC790, SUBMARINE SIMULATION, LONGITUDINAL FREEDOM

1. DESCRIPTION

This program calculates the dynamic changes in submarine position, velocity, and attitude as a function of time for longitudinal freedom. It is very similar to Program EB920, except that lateral freedom (rudder change with attendent roll and turn) is not provided for. Therefore, only longitudinal runs due to thrust or elevator changes can be provided with this program.

The purpose in preparing this program, in view of the existence of EB920 (all degrees of freedom), is the shortened computer running time for the abbreviated program. This will be a real advantage where a large number of meander, overshoot, or acceleration runs are required. Figure 8 is a block diagram, showing the general layout of the program and the subroutines used. Equation (9) through (12) give the mathematical model used. It follows "Standard Equations of Motion for Submarine Limitations" except that all lateral coefficients have been removed.

The program is written in basic FORTRAN IV for use on any digital computer with a FORTRAN compiler. It occupied 6K words when run is on IBM 360/40 computer. Program results are printed as a function of time and individual parameters can be plotted if associated CALCOMP plotter software is available.

The program will accept values of initial conditions for control surface positions, speed (and components) attitude and depth. This allows the submarine to be placed in a steady state dive, level flight, etc., at time zero or start of the computer run. These inputs are:

UC	-	command speed
DS	-	stern elevator position
DB	-	bow elevator position
W	-	component of velocity in the z-direction
Q	-	angular acceleration component about the y-axis relative to fluid.
THETA	-	angle of pitch
Z	-	depth

This program will give identical results as the longitudinal channel of EB920 when the same coefficients are used but the running time is one-eighth and the space required is one-third.

2. SUBROUTINE DESCRIPTIONS

The subroutines utilized by this program are shown in block diagram form in figure 8. A brief description of each subroutine is included for clarification of operation of the total program.

a. INPUT - This subroutine reads a data deck specifying program options, control, and initial dynamic conditions and the particular

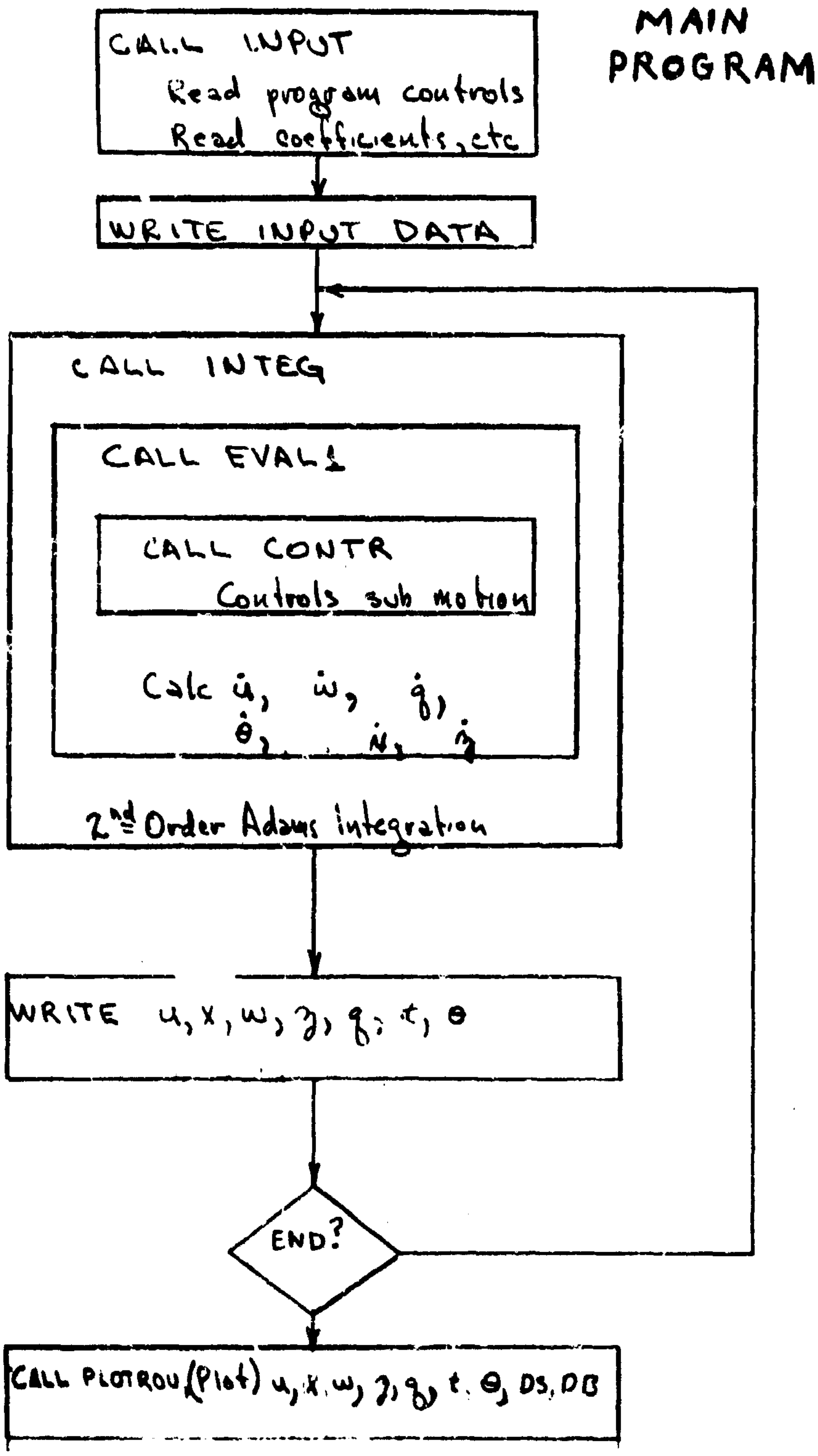


Figure 8. CC790 Submarine Simulation Program, Block Diagram

Equations of Motion for Longitudinal Freedom Only (ZC790 Math Model)

AXIAL FORCE (8)

$$
\begin{aligned}
m[\dot{u} + wq - X_a q^2 + Z_a \dot{q}] = \\
&+ \frac{\rho}{2} \ell^4 [X_{qq} q^2] \\
&+ \frac{\rho}{2} \ell^3 [X_{\dot{u}} \dot{u} + X_{wq} wq] \\
&+ \frac{\rho}{2} \ell^2 [X_{uu} u^2 + X_{ww} w^2] \\
&+ \frac{\rho}{2} \ell^2 u^2 [X_{\delta s \delta s} \delta s^2 + X_{\delta b \delta b} \delta b^2] \\
&+ \frac{\rho}{2} \ell^2 [a_i u^2 + b_i u u_c + c_i u_c^2] \\
&- (W - B) \sin\theta \\
&+ \frac{\rho}{2} \ell^2 [X_{ww\eta} w^2 + X_{\delta s \delta s \eta} \delta_s^2 u^2](\eta - 1)
\end{aligned}
$$

KINEMATIC RELATIONS (9)

$$U^2 = u^2 + w^2$$

$$\dot{\theta} = q$$

$$\dot{Z}_o = w \cos\theta - u \sin\theta$$

$$\dot{X}_o = u \cos\theta + w \sin\theta$$

PITCHING MOMENT (10)

$$
\begin{aligned}
I_y \dot{q} &+ m[Z_G(\dot{u} + wq) - X_G(\dot{w} - uq)] = \\
&\frac{\rho}{2} \ell^5 [M_{\dot{q}} \dot{q} + M_{q|q|} q|q|] \\
&+ \frac{\rho}{2} \ell^4 [M_{\dot{w}} \dot{w}] \\
&+ \frac{\rho}{2} \ell^4 [M_q uq + M_{|q|\delta_s} u|q|\delta_s + M_{|w|q} |w| q] \\
&+ \frac{\rho}{2} \ell^3 [M_* u^2 + M_w uw + M_{w|w|} w|w|] \\
&+ \frac{\rho}{2} \ell^3 [M_{|w|} u|w| + M_{ww} |w|w|w||] \\
&+ \frac{\rho}{2} \ell^3 [M_{\delta_s} u^2 \delta_s + M_{\delta_b} u^2 \delta_b] \\
&- (X_G W - X_B B) \cos\theta - (Z_G W - Z_B B) \sin\theta \\
&+ \frac{\rho}{2} \ell^4 M_{q\eta} uq(\eta - 1) \\
&+ \frac{\rho}{2} \ell^3 [M_{w\eta} uw + M_{w|w|\eta} w|w| + M_{\delta_s\eta} \delta_s u^2](\eta - 1)
\end{aligned}
$$

NORMAL FORCE (11)

$$m\left[\dot{w} - uq - z_G q^2 - x_G \dot{q}\right] =$$

$$+ \frac{\rho}{2} \ell^4 \left[Z_{\dot{q}} \dot{q} \right]$$

$$+ \frac{\rho}{2} \ell^3 \left[Z_{\dot{w}} \right]$$

$$+ \frac{\rho}{2} \ell^3 \left[Z_q uq + Z_{|q|\delta s} u|q|\delta_s + Z_{w|q|} w|q| \right]$$

$$+ \frac{\rho}{2} \ell^2 \left[Z_* u^2 + Z_w uw + Z_{w|w|} w|w| \right]$$

$$+ \frac{\rho}{2} \ell^2 \left[Z_{|w|} u|w| + Z_{ww} |w|w|| \right]$$

$$+ \frac{\rho}{2} \ell^2 \left[Z_{\delta s} u^2 \delta_s + Z_{\delta b} u^2 \delta_b \right]$$

$$+ (W - B) \cos\theta$$

$$+ \frac{\rho}{2} \ell^3 Z_{q\eta} uq (\eta - 1)$$

$$+ \frac{\rho}{2} \ell^2 \left[Z_{w\eta} uw + Z_{w|w|\eta} w|w| + Z_{\delta s\eta} \delta_s u^2 \right] (\eta - 1)$$

coefficients and constants for the submarine to be simulated. It provides for holding initial values of all variables for use on a subsequent run. It provides for reading additional cards for successive runs, and finally for exit from the program on a blank card.

b. INTEG - The only integration algorithm available is 2nd order Adams. This program is used mainly for varification runs rather than research so different integration methods are not required.

c. EVALI - This subroutine evaluates the equations of motion in accordance with the mathematical model given in section II. B. 1. It is identical to the equations used in program ZB920 except that the lateral, roll, and yaw channels have been removed and all longitudinal coefficients using v, p, and r have been set to zero.

d. CONTR - The block diagram of figure 9 outlines the CONTR subroutine. This subroutine allows selection of one of several preplanned maneuvers that are used most often in submarine research studies. The operation is identical to that of the CONTR subrouting in program ZB920 except that provisions for moving the rudder have been removed. The variable NS will set the desired bow or sternplane schedule as follows:

NS	= 0	Constant input value
NS	= 1	Meander or Overshoot (figure 3)
NS	= 2	DS - Impulse
NS	= 3	Acceleration/Deceleration
NS	= 4	Maximum Acceleration/Deceleration

A useful addition of the program for elevator-impulse control is the computer output of punched cards of the submarine pitch angle versus time. These are punched in 2E15.7 format (THETA, TIME) from T_0 to TLIM. The cards are useful in other programs associated with submarine response and described elsewhere in this report. Most associated programs are based on cards punched at two-second intervals. To achieve this, the input value of integration interval H, must be 0.25 seconds and NPNT must be eight.

The integration in ZC 790 is performed with the second-order Adams method. The results may be slightly different from integration to that performed by the Runge-Kutta method in the EB920 program. This difference is magnified for rapidly changing conditions as prevail in impulse runs. Therefore, care should be exercised in the use of impulse runs for this program, and comparison made with the EB920 program. Greatest accuracy will accrue by the use of the Runge-Kutta integration of EB920.

The autopilot control of this program uses the identical equation (8) used by the autopilot control in program EB920.

e. PLOTROU - This is identical to PLOTROU of EB920 except that the variables v, p, r, PHI, PSI, y, and DR are not saved or plotted.

3. INPUT DATA DECK

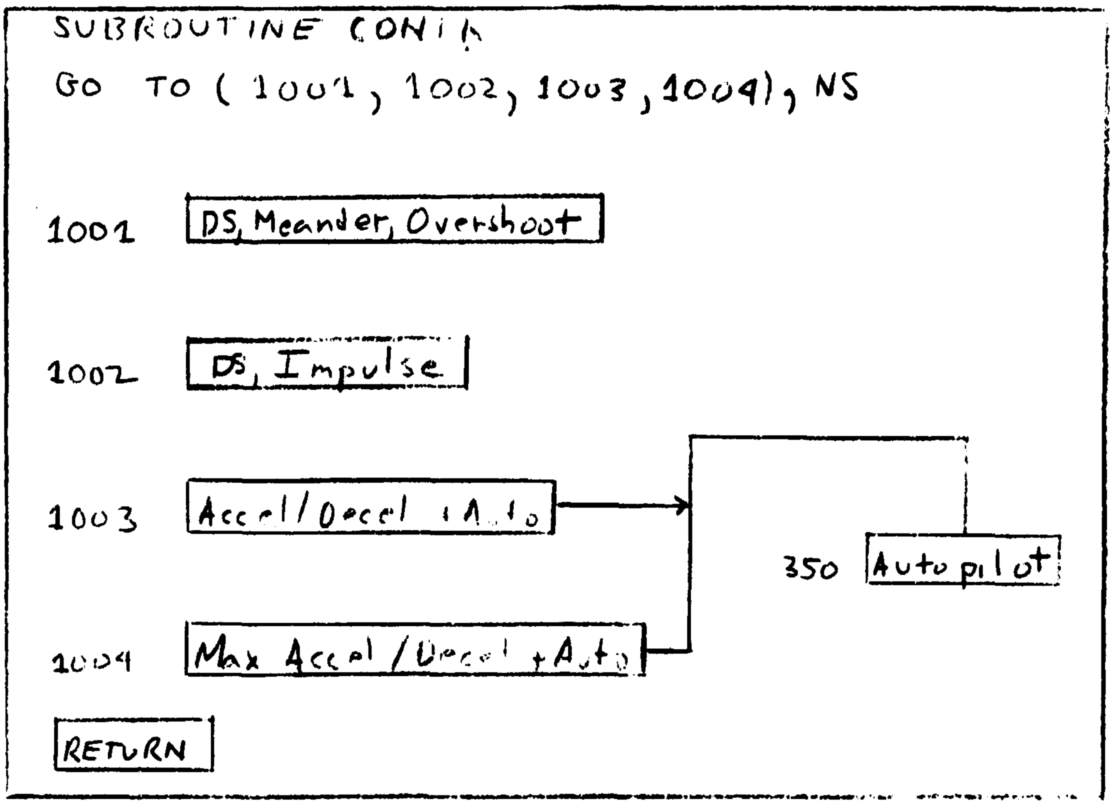


Figure 9. Subroutine CONTR, ZC790, Block Diagram

This section describes the input data deck, defines the FORTRAN variables in terms of the equations of motion defined in the mathematical model, and shows how to input data to perform various kinds of simulated submarine operations. The coefficients referred to here are those included in the reference "Standard Equations of Motion for Submarine Simulation". The coefficients are restricted to those in X, Z, and M, corresponding to those used in calculating axial force, normal force, and pitching moment on the submarine. The coefficients in Y, K, and N are not used because they affect only lateral forces and motion of the submarine. The input deck should follow table 6 exactly.

TABLE 6. INPUT DATA DECK, PROGRAM ZC790

Card	Column(s)	Format	Description
Control Flags			
1	1-5	I5	NPNT. Data will print out at T_0 and each NPNT = integration step, each 2 sec if NPNT = 8 and H = .25 seconds. Right adjust all integer values.
1	6-10	If	IPLOT. To exercise plot option, IPLOT = 1. Leave blank

TABLE 6. INPUT DATA DECK, PROGRAM ZC790 (cont.)

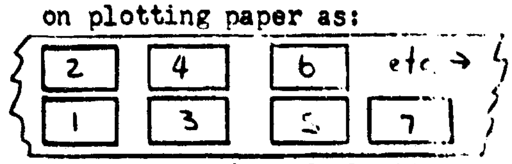
Card	Column(s)	Format	Description
			(zero) for no plots.
1	11-15	I5	IRUN. Identification number for individual runs. If IRUN = 0 a normal exit is made.
1	16-20	I5	NPLT. Data will plot at T_0 each NPLTth integration step.
1	21-25	I5	IOPT. This option will allow changing any one or more of the input values for a succeeding run without putting in all the other values. IOPT = 1 to exercise ; blank for no selection. Another run may still follow, for IOPT = Ø, but all input data cards must be read again. More information on this variable is included at card 19.
1	26-30	I5	NS. This variable selects type of submarine control in CONTR subroutine NS = 0, Fixed control per initial conditions NS = 1, Overshoot, meander, ect. NS = 2, DS-impulse NS = 3, Acceleration/deceleration NS = 4, Maximum acceleration/ deceleration
Plot card			Card must be here (blank) even if no plots are required.
2	1-35	7I5	ILOC(I) I = 2,8. Value of I defines which variable will be plotted against time. Sequence defines order of plot on plotting paper as:  order of plotting Plot variable: ILOC(2) - U, component of velocity in the X-direction, feet/second

TABLE 6. INPUT DATA DECK, PROGRAM ZC790 (cont.)

Card	Column(s)	Format	Description
			ILOC(3) - W, component of velocity in the z-direction, feet/second ILOC(4) - Q, angular velocity about y-axis, radians/second ILOC(5) - theta (Θ), pitch angle, radians ILOC(6) - Z, depth, feet ILOC(7) - DS, stern plane position, degrees ILOC(8) - DB, bow plane position, degrees
Timing			
3	1-10	F10.5	TO, starting time, seconds
3	11-20	F10.5	HO, integration time increment, seconds
3	21-30	F10.5	TLIM, time of run, seconds
Initial conditions			
4	1-10	F10.5	Y(1), U, velocity component in the X-direction, feet/second
4	11-20	F10.5	Y(2), W, velocity component in z-direction, feet/second
4	21-30	F10.5	Y(3), Q, angular velocity about, y-axis, radians/second
4	31-40	F10.5	Y(4), THETA(Θ), pitch angle, radians
4	41-50	F10.5	Y(5), Z, depth, feet
Coefficient Cards			
5	1-80	8F10.5	XQQ, XUD, XWQ, XUU, XWW, XDSDS, XDBDB, XWWE
6	1-10	F10.5	XDSDSE
7	1-80	8F10.5	ZQD, ZWD, ZQ, ZAQDS, ZWAQ, ZSTR, ZW, ZWAW

TABLE 6. INPUT DATA DECK, PROGRAM ZC790 (cont.)

Card	Column(s)	Format	Description
8	1-80	8F10.5	ZAW, ZWW, ZDS, ZDB, ZQE, ZWE, ZWAWE, ZDSE
9	1-80	8F10.5	AMQD, AMQAQ, AMWD, AMQ, AMAQDS, AMAWQ, AMSTR, AMW
10	1-80	8F10.5	AMWAW, AMAW, AMWW, AMDS, AMDB, AMQE, AMWE, AMAWE
11	1-80	F10.5	AMDSE
Submarine Constants			
12	1-10	F10.5	IY, moment of inertia about the y-axis, slug/ft^2
13	1-10	F10.5	CW, weight including water in free-flooding space, pounds
13	11-20	F10.5	CB, buoyancy, pounds
13	21-30	F10.5	UC, initial command speed, feet/second
13	31-40	F10.5	XB, x-component of center of buoyancy, feet
13	41-50	F10.5	ZB, z-component of center of buoyancy, feet
14	1-10	F10.5	DS, initial value of stern elevator position, radians
14	11-20	F10.5	DB, initial value of bow elevator position, radians
14	21-30	F10.5	RHO, density of sea water, slugs/feet^3
14	31-40	F10.5	AL, submarine length, feet
14	41-50	F10.5	AM, submarine mass, including water in free flooding space, slugs
15	1-10	F10.5	ETAHI(η-high), upper reference value of UC/UMAG, dimensionless
15	11-20	F10.5	ETALO(η-low), lower reference value of UC/UMAG, dimensionless

TABLE 6. INPUT DATA DECK, PROGRAM ZC790 (cont.)

Card	Column(s)	Format	Description
15	21-30	F10.5	A11, value of a_i for ETA ETAHI
15	31-40	F10.5	A12, value of b_i for ETA ETAHI
15	41-50	F10.5	A13, value of c_i for ETA ETAHI
16	1-10	F10.5	A21, value of a_i for ETALO ETA ETAHI
16	11-20	F10.5	A22, value of b_i for ETALO ETA ETAHI
16	21-30	F10.5	A23, value of c_i for ETALO ETA ETAHI
16	31-40	F10.5	A31, value of a_i for ETA ETALO
16	41-50	F10.5	A32, value of b_i for ETA ETALO
16	51-60	F10.5	A33, value of c_i for ETA ETALO
17	1-10	F10.5	XG, x-component of center of gravity, feet
17	11-20	F10.5	ZG, z-component of center of gravity, feet
Surface Control Schedule			
18	1-10	F10.5	TIME, various periods of time depending on NS in control routine, seconds NS — Time 0 — Not used 1 — Initial steady state period (See figure 3) 2 — Not used in input 3 — Period of constant application of each command speed 4 — Period of constant application of each command speed
18	11-20	F10.5	R1, negative rate of change of elevator position, radians/second during meander runs (NS = 1), radians/second (See figure 3)

TABLE 6. INPUT DATA DECK, PROGRAM ZC790 (cont.)

Card	Column(s)	Format	Description
18	21-30	F10.5	DELTMA, maximum negative swing of elevator (must be more negative than DS, initial condition) during meander runs (NS = 1), radians (See figure 3)
18	31-40	F10.5	SWMAX (Θ'), execute pitch angle, maximum submarine dive angle during meander type run (NS = 1) before turning elevators to a more positive value, radians
18	41-50	F10.5	R2, positive rate of change of elevator position after submarine reaches Θ', radians/second (See figure 3)
18	51-60	F10.5	DELTMI, maximum position to which elevators are moved during 'R2' change, radians (See figure 3)
18	61-70	F10.5	DSF, final value of elevator position during DS - impulse run (NS = 2), radians.
Additional Run Controls			If IOPT was "0" on card 1 of the last data deck, card 32 can be either blank for a normal exit or card 1 of a new data deck for additional runs. If IOPT = F on card 1 of the last data deck, see below.
19	1-5	I5	IRUN, run numbers IRUN = 0 (blank card) a new data deck is read. A blank card at the start of a data deck results in a normal exit so two blank cards at the end will always end the run. IRUN = Integer. Use as new run number and continue reading cards below.
20	1-5	I5	NDEX, common location for parameter to be changed. Table 8 shows the variable names versus the the common

TABLE 6. INPUT DATA DECK, PROGRAM ZC790 (cont.)

Card	Column(s)	Format	Description
			locations, NDEX
20	11-20	F10.5	VALUE, new value of parameter changed in columns 1-5 of this card.
20 + n	1-20	I5, 5X, 10.5	Repeat card 20 as desired.
20 + n + 1	-	Blank	Start new run after all changes have been made

4. OUTPUT DATA

a. Printout - This program will print out titled, numbered pages. The first page provides the input data as shown in figure 10. Subsequent pages provide the variables noted in table 7.

TABLE 7. OUTPUT VARIABLES, PROGRAM ZC790

Variable	Format	Units	Description
U	E13.6	feet/second	Velocity component in the x-direction
W	E13.6	feet/second	Velocity component in the z-direction
Q	E13.6	radians/seconds	Angular velocity about the y-axis
THETA	E13.6	radians	Submarine pitch angle
Z	E13.6	feet	Submarine depth
T	E13.6	seconds	Time

A sample data sheet is shown in figure 11. (Coefficients used in this run are trial or synthetic coefficients). This data is a function of the time of the run, T. The frequency of printout is a function of NPNT, printed each NPNTth integration cycle. For instance:

For the integration which used ICYC = 1. If H, integration time period is 0.25 second and NPNT = 8

$$T* = \frac{NPNT \times H}{ICYC} = \frac{8 \times 0.25}{1} \quad 2 \text{ seconds} \qquad (12)$$

This printout would be at t = 0 and each two seconds thereafter.

TABLE 8. ZC 790. COMMON LAYOUT

Location	Term	Location	Term	Location	Term	Location	Term
1	H	37	ZWAWE	73	A22		
2	N	38	ZDSE	74	A23		
3	IS1	39	AMQD	75	A31		
4	NPNT	40	AMQAQ	76	A32		
5	TLIM	41	AMWD	77	A33		
6	RHOL2	42	AMQ	78	XG		
7	RHOL3	43	AMAQDS	79	ZG		
8	RHOL4	44	AMAWQ	80	ILOC(1)		
9	RHOL5	45	AMSTR	81	(2)		
10	WMB	46	AMW	82	(3)		
11	ETA	47	AMWAW	83	(4)		
12	ETAM1	48	AMAW	84	(5)		
13	ISW2	49	AMWW	85	(6)		
14	XQQ	50	AMDS	86	(7)		
15	XUD	51	AMDB	87	(8)		
16	XWQ	52	AMQE	88	IPLOT		
17	XUU	53	AMWE	89	IRUN		
18	XWW	54	AMAWE	90	IOPEN		
19	XDSDS	55	AMDSE	91	NPLT		
20	XDBDB	56	IY	92	IOPT		
21	XWWE	57	CW	93	Y(1)		
22	XDSDSE	58	CB	94	(2)		
23	ZQD	59	UC	95	(3)		
24	ZWD	60	XB	96	(4)		
25	ZQ	61	ZB	97	(5)		
26	ZAQDS	62	DS	98	(6)		
27	ZWAQ	63	DB	99	TIME		
28	ZSTR	64	RHO	100	R1		
29	ZW	65	AL	101	DELTMA		
30	ZWAW	66	AM	102	SWMAX		
31	ZAW	67	ETAHI	103	R2		
32	ZWW	68	ETALO	104	DELTMI		
33	ZDS	69	A11	105	DSF		
34	ZDB	70	A12	106	NS		
35	ZQE	71	A13				
36	ZWE	72	A21				

```
ZC790              SUBMARINE SIMULATION, LONGITUDINAL FREEDOM                                   PAGE   1

XQQ     -0.10000E-03        ZQQ     -0.10000E-03        MQQ     -0.70000E-03
XUQ     -0.40000E-03        ZWQ     -0.14000E-01        MQAQ     0.0
XWQ     -0.14000E-01        ZQ      -0.70000E-02        MWQ     -0.13000E-03
XUU      0.0                ZAQDS    0.30000E-02        MQ      -0.40000E-02
XWW      0.50000E-02        ZWAQ    -0.10000E-01        MAQDS   -0.50000E-03
XDSDS   -0.30000E-02        ZSTR     0.22000E-03        MAWQ    -0.30000E-02
XDRDR   -0.50000E-02        ZW      -0.17000E-01        MSTR     0.75000E-04
XWF      0.0                ZWAW    -0.55000E-01        MW       0.75000E-02
XDSDSF   0.0                ZAW      0.0                MWAW    -0.14000E-01
                            ZWW      0.0                MAW      0.0
                            ZDS     -0.85000E-02        MWW      0.0
                            ZDB     -0.50000E-02        MDS     -0.40000E-02
                            ZQF      0.0                MDB      0.10000E-02
                            ZWF      0.0                MQF      0.0
                            ZWAWF    0.0                MWF      0.0
                            ZDSF     0.0                MWAWF    0.0
                                                        MDSF     0.0

                 IY     0.13000E 10
W    0.10000E 08 B      0.10000E 08  UC     0.84450E 01  XB    0.0                         ZB     0.0
                 DS     0.15254E-01  DR     0.0          RHO   0.20000E 01 L      0.27500E 03  M      0.30000E 06
A11 -0.15000E-02 A21   -0.29500E-02  A31   -0.36400E-02                    ETAHI  0.50000E 00  ETALO -0.50000E 00
A12 -0.13000E-02 A22    0.12000E-02  A32   -0.39200E-02  XG    0.0                         ZG     0.0
A13  0.28000E-02 A23    0.0          A33   -0.30850E-02  H     0.25000E 00
TIME 0.50000E 01 R1    -0.97270E-01  DELTMA -0.34906E 00 SWMAX 0.13963E 00 R2     0.97270E-01  DELTMI 0.37957E 00
DSF  0.0                             ICYC   1            NS    1

RUN NO   1
```

Figure 10.   Input Variable Format, ZC790

ZC790 SUBMARINE SIMULATION, LONGITUDINAL FREEDOM PAGE 2

U	W	Q	THETA	Z	T
0.844500E 01	0.435340E-01	0.0	0.515590E-02	0.800000E 03	0.0
0.844493E 01	0.440067E-01	0.113519E-03	0.540057E-02	0.799996E 03	0.400000E 01
0.844094E 01	0.813849E-01	0.124640E-02	0.715393E-02	0.800011E 03	0.800000E 01
0.841336E 01	0.202648E 00	0.423130E-02	0.181783E-01	0.800176E 03	0.120000E 02
0.837746E 01	0.334770E 00	0.667359E-02	0.401781E-01	0.800294E 03	0.160000E 02
0.833118E 01	0.468018E 00	0.862067E-02	0.709152E-01	0.800064E 03	0.200000E 02
0.827761E 01	0.597288E 00	0.101749E-01	0.108625E 00	0.799227E 03	0.240000E 02
0.820335E 01	0.714805E 00	0.112540E-01	0.151836E 00	0.797571E 03	0.280000E 02
0.815029E 01	0.754215E 00	0.983829E-02	0.194982E 00	0.794940E 03	0.320000E 02
0.810281E 01	0.689551E 00	0.611240E-02	0.227403E 00	0.790839E 03	0.360000E 02
0.807050E 01	0.578499E 00	0.268664E-02	0.244646E 00	0.785710E 03	0.400000E 02
0.805433E 01	0.453518E 00	0.113629E-03	0.250014E 00	0.779791E 03	0.440000E 02
0.804623E 01	0.321470E 00	-0.192544E-02	0.246254E 00	0.773363E 03	0.480000E 02
0.804061E 01	0.187149E 00	-0.364274E-02	0.235027E 00	0.766663E 03	0.520000E 02
0.803351E 01	0.523993E-01	-0.514855E-02	0.217387E 00	0.759902E 03	0.560000E 02

Figure 11. Output Variable Format, ZC790

b. Plots - The program can optionally plot any of the printed values. In addition, the control parameter DS and DB (plane position, radians) can also be plotted. This output is identical to that of figure 7.

5. LISTINGS AND FLOW CHARTS

Appendix A contains the listings and flow chart for this program.

C. EC 470, INITIAL CONDITION COMPUTATION FOR SIMULATION

1. DESCRIPTION

In order to evaluate longitudinal performance of the submarine, a set of initial conditions for level flight is required. These initial conditions are a set of neutral angles defined as Θ, steady state pitch angle; DS, sternplane angle; and DB, bow (sail) plane angle at a particular speed. This requires that the angle of attack equal the pitch angle and that the accelerations $\dot{u}$, $\dot{w}$, and $\dot{q}$ equal zero. Reference is made to angle of attack in many texts, including the NSRDC reports but the parameter does not appear in the equations of motion. This is because a body axis system of coordinates is used instead of a flight path set. The angle of attack is the angle between the flight path and the body when roll angle is zero.

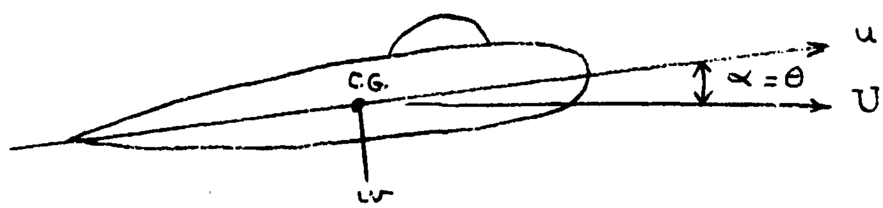


Figure 12. Angle of Attack.

The angle of attack (and in the case of steady level flight, the pitch angle) can be calculated from the relationship

$$U^2 = u^2 + w^2 \tag{13}$$

$$\sin \Theta = \sin \alpha = \frac{w}{U} = \frac{w}{\sqrt{u^2 + w^2}} \tag{14}$$

as seen in figure 12. This removes the parameter Θ by replacing it with a function of w. For any speed, U, the equations of motion can be solved for level flight, in terms of U and δ_s or δ_b, by setting $\dot{w}$, q, $\dot{q}$, and all lateral terms to zero. Equation (3) is determined from Normal Equation (3) of the NSRDC Standard Equations and

$$Z_{w|w|}\, w|w| + Z_{ww}\, w^2 + Z_w\, uw + Z_{|w|}\, u|w| + Z_* u^2 + Z_\delta u^2 \delta = 0 \tag{15}$$

equation from the pitching moment equation (5).

$$M_{w|w|}\, w|w| + M_{ww}\, w^2 + M_w\, uw + M_{|w|}\, u|w|$$
$$+ M_* u^2 + M_\delta u^2 \delta + \frac{2B\, Z_B\, w}{\rho l^3 \sqrt{u^2 + w^2}} = 0 \qquad (16)$$

ρ = density of water, l = length of submarine, and δ can be either δ_S or δ_B provided that the correct coefficients $Z_{\delta S}$, $M_{\delta S}$ or $Z_{\delta B}$, $M_{\delta B}$ are used as inputs.

This program solves Equations (15) and (16) for w and δ. The program utilizes all coefficients of interest so that it can be used with any set of coefficients applicable to the NSRDC equations. The pitch equation is solved for δ with a trial value of w, and this value is used in the normal equation via Newton's method. Theta is then calculated by equation (17).

$$\Theta = \tan^{-1} w/u \text{ radians} \qquad (17)$$

This program serves another purpose in addition to determining the initial conditions for longitudinal runs. The values for δ are determined for the operational speed range of the submarine and at some point the values become very large. This speed is known as the critical speed because at this speed the controls are ineffective in controlling the pitch attitude of the submarine. The critical speed points determined by this program can be checked against the data sources to insure effective training through the simulation.

2. INPUT DATA DECK

The inputs to the program include the appropriate submarine coefficients, including choices of ZDS or ZDB and MDS or MDB for stern or sailplanes respectively. The other plane is considered to be set at zero and has no effect on the submarine trim. The submarine physical constants, and the speed of operation are needed also. These values are identified in detail with their locations on punch cards in table 9.

TABLE 9. INPUT DATA DECK, PROGRAM EC 470.

Card	Column(s)	Format	Description
1	1-50	5F10.5	ZWAW, ZWW, ZW, ZAW, ZSTR
1	51-60	F10.5	ZDEL, use either ZDS or ZDB as desired.
2	1-50	5F10.5	MWAW, MWW, MW, MAW, MSTR
2	51-60	F10.5	MDEL, use either MDS or

TABLE 9. INPUT DATA DECK, PROGRAM EC470 (cont.)

Card	Column(s)	Format	Description
			MDB in accord with selection of ZDEL above.
3	1-10	F10.5	B, buoyancy, pounds
3	11-20	F10.5	AB, z-component of center of buoyancy location, feet
3	21-30	F10.5	RHO, , density of sea water, slugs/ft^3
3	31-40	F10.5	AL, , submarine length, feet
4	1-10	F10.5	U, submarine forward velocity, feet/second. This should be highest speed desired.
4	11-20	F10.5	WZERO, , trial value of as required in solution of problem. This initial guess should be close; 1st approximation is .002 U radians (U in feet/second).
4	21-30	F10.5	ULIM, submarine forward velocity, feet/second. This is the slowest speed desired. The program will calculate initial conditions from U thru ULIM in one (1) knot (1.689 feet/second) intervals.
5	-	-	Blank card for normal end of job

3. OUTPUT DATA

Figure 13 is a sample output data sheet using trial or synthetic coefficients. The output is presented in rows of data at one knot intervals in speed from the highest requested speed, U to the lowest, ULIM. The following parameters are printed:

a. Input data - All coefficients and submarine physical constants as provided in the first three input data cards. Units are same as those for the input data.

b. Output data

U - forward speed, feet/second
U - forward speed, knots
W - Normal speed, feet/second

EC470 - INITIAL CONDITIONS

ZAAL	ZAB	ZB	ZAW	ZSTE	ZDEL
-0.550000E-01	0.0	-0.170000E-01	0.0	0.220000E-02	-0.850000E-02
XAAL	XAB	XB	XAW	XSTE	XDEL
-0.100000E-01	0.0	0.750000E-02	0.0	0.750000E-04	-0.400000E-02
[illegible]	ZA	RHO	I		
0.130000E 00	-0.160000E 01	0.260000E 01	0.275000E 03		

U(FT/SEC)	U(KTS)	W(FT/SEC)	DEL(RAD)	THETA(RAD)	DEL(DEG)	THETA(DEG)
0.622250E 02	0.250000E 02	0.759242E-01	0.221782E-01	0.199069E-02	0.126600E 01	0.113317E 00
0.405360E 02	0.240000E 02	0.748197E-01	0.221699E-01	0.195510E-02	0.126446E 01	0.113531E 00
0.398570E 02	0.230000E 02	0.733226E-01	0.220593E-01	0.190734E-02	0.126385E 01	0.113231E 00
0.371540E 02	0.220000E 02	0.704361E-01	0.220441E-01	0.190435E-02	0.126315E 01	0.109224E 00
0.354680E 02	0.210000E 02	0.674523E-01	0.220321E-01	0.191329E-02	0.126235E 01	0.112623E 00
0.337730E 02	0.200000E 02	0.644331E-01	0.220159E-01	0.192135E-02	0.126141E 01	0.111334E 00
0.320090E 02	0.190000E 02	0.614615E-01	0.219566E-01	0.193381E-02	0.126031E 01	0.111627E 00
0.304010E 02	0.180000E 02	0.580726E-01	0.218730E-01	0.194700E-02	0.125301E 01	0.111259E 00
0.287120E 02	0.170000E 02	0.571157E-01	0.219468E-01	0.195540E-02	0.125744E 01	0.112234E 00
0.270230E 02	0.160000E 02	0.529011E-01	0.219119E-01	0.197163E-02	0.125558E 01	0.112354E 00
0.253340E 02	0.150000E 02	[illegible]	0.218734E-01	0.199157E-02	0.125324E 01	0.114103E 00
0.236450E 02	0.140000E 02	0.476319E-01	0.218731E-01	0.201649E-02	0.125127E 01	0.115537E 00
0.219560E 02	0.130000E 02	0.449716E-01	0.217532E-01	0.204327E-02	0.124478E 01	0.117357E 00
0.202670E 02	0.120000E 02	0.423550E-01	0.216746E-01	0.208974E-02	0.124186E 01	0.119734E 00
0.185790E 02	0.110000E 02	0.398630E-01	0.215414E-01	0.214561E-02	0.123530E 01	0.122734E 00
0.168980E 02	0.100000E 02	0.375580E-01	0.214929E-01	0.222373E-02	0.122643E 01	0.127610E 00
0.152008E 02	0.900000E 01	0.355319E-01	0.211493E-01	0.233381E-02	0.121291E 01	0.134134E 00
0.135114E 02	0.800000E 01	0.341555E-01	0.207299E-01	0.252115E-02	0.119169E 01	0.144451E 00
0.118225E 02	0.700000E 01	0.336906E-01	0.201411E-01	0.284445E-02	0.115400E 01	0.162975E 00
0.101335E 02	0.600000E 01	0.350121E-01	0.197136E-01	0.354174E-02	0.117221E 01	0.202936E 00
0.844480E 01	0.500000E 01	0.517093E-01	0.137322E-01	0.506455E-02	[illegible]	0.341734E 00

Figure 13. Output Variable Format, EC470

DEL - DS or DB, depending in the selection of the coefficients, ADS and MDS or ZDB and MDB respectively, radians.
THETA - Θ, submarine pitch angle, radians
DEL - DS or DB, degrees
THETA - Θ, degrees

4. LISTINGS AND FLOW CHART

The listings and flow chart for program EC470 are included in appendix A.

D. EC430, CENTER OF GRAVITY COMPUTATION

1. DESCRIPTION

This program calculates the new location of the center of gravity, in three planes, due to the flooding of any combination of tanks on a submarine.

The program solves equations stated below, and points out the new total weight, three components of center of gravity position, and a record of the tanks filled.

$$X_G = \frac{W_1 X_1 + \sum W_i X_i}{W_1 + \sum W_i}$$

$$Y_G = \frac{W_1 Y_1 + \sum W_i Y_i}{W_1 + \sum W_i}$$

$$Z_G = \frac{W_1 Z_1 + \sum W_i Z_i}{W_1 + \sum W_i}$$

where X_G = x - component of center of gravity location, feet

Y_G = y - component of center of gravity location, feet

Z_G = z - component of center of gravity location, feet

W_1 = weight of submarine, pounds

X_1 = x - component of center of gravity basic submarine, feet

Y_1 = y - component of center of gravity basic submarine, feet

Z_1 = z - component of center of gravity basic submarine, feet

W_i = weight of water in i^{th} tank, pounds

X_i = x - component of center of gravity of i^{th} tank, feet

Y_i = y - component of center of gravity of i^{th} tank, feet

Z_i = z - component of center of gravity of i^{th} tank, feet

W = $Wo + \sum Wi$, pounds

i = integer from 2 to 50, depending on number of tanks considered

2. INPUT DATA DECK

The input variables to the program include the numbers of particular

tanks, weight of water when full, three components of center of gravity of each tank when full, and particular control variables. These inputs are described in detail with their locations on punch cards in table 10.

TABLE 10. INPUT DATA DECK, PROGRAM EC430

Card	Column(s)	Format	Description
1	1-5	I5	N, number of tanks (including one (1) for submarine, considered as a tank)
1	6-10	I5	IPNT, controls printant data IPNT = 1 - Input data printed IPNT = 0 - Input data not printed
2	1-10	F10.5	WI, Wi, weight of water in tanks, pounds. First weight, w_1, is weight of submarine
2	11-20	F10.5	XI, Wi, x-component of center of gravity, pounds. First value, X_1, is for submarine (usually = 0).
2	21-30	F10.5	YI, Yi, y-component of center of gravity, pounds. First value, Y_1, is for submarine (usually = 0).
2	31-40	F10.5	ZI, Zi, z-component of center of gravity, pounds. First value, Z_1, is for submarine (usually = 0).
2	41-50	F10.5	WI, Wi = W(2), weight of water in #2 tank.
2	51-60	F10.5	XI, Xi = X(2), x-component of of C.G. for #2 tank
2	61-70	F10.5	YI, Yi = Y(2), y-component of C.G. for #2 tank
2	71-80	F10.5	ZI, Zi = Z(2), z-component of C.G. for #2 tank
3-n	1-80	8F10.5	Repeat card number two (2) until N weights and C.G. components have been entered, one set for each tank. Maximum number of tanks (sets) is

TABLE 10. INPUT DATA DECK, PROGRAM EC430 (cont.)

Card	Column(s)	Format	Description
			fifty (50).
n+1	1-n	nI1	ICTL(I) I = 1,N This single array represents all tanks to be considered, #1 (the submarine itself) thru N (the largest number of tanks up thru 50). ICTL(I) = 0, i^{th} tank is empty. ICTL(I) = 1, i^{th} tank is full and the W, X, Y, and Z values of this tank will be included in the calculations.
M+2-m			Stack as many of these cards as desired, one card for each run
m+1	1	I1	9, this is normal end of job card.

3. OUTPUT DATA

A sample of the output data sheets is shown in figure 14. This sample is in two pages, numbered sequentially and each identified by the program number and title.

Page one shows the input data, N, W, X, Y, and Z, number of the tank, weight of water of the tank and the x, y, and z coordinates of the center of gravity of each full tank.

Page two contains two sets of data. The first includes a listing of all tanks considered from 1 thru N. Below each tank number is a second digit which is "1" if the tank is filled and the weights and moment arms for that tank are considered in the calculations. This number is "0" if the tank is empty.

4. LISTINGS AND FLOW CHART

The listings and flow chart for program EC430 are included in appendix A.

EC430 COMPUTED CENTER OF GRAVITY PAGE 1

[illegible]	[illegible]	[illegible]	Y	Z
1	[illegible]	[illegible]	0.0	0.0
2	[illegible]	[illegible]	0.0	-0.200000E 01
3	[illegible]	[illegible]	0.0	-0.200000E 01
4	[illegible]	[illegible]	0.0	-0.200000E 01
5	[illegible]	[illegible]	0.0	0.400000E 01
6	[illegible]	[illegible]	0.0	0.200000E 01
7	[illegible]	[illegible]	0.0	0.900000E 01
8	[illegible]	[illegible]	0.0	0.300000E 01
9	[illegible]	[illegible]	0.0	0.500000E 02
10	[illegible]	[illegible]	0.0	-0.370000E 02
11	[illegible]	[illegible]	0.0	0.0
12	[illegible]	[illegible]	0.0	0.0
13	[illegible]	[illegible]	0.0	0.500000E 02
14	[illegible]	[illegible]	0.0	0.400000E 01
15	[illegible]	[illegible]	0.0	0.200000E 01

Figure 14. Output Variable Format, EC430

```
  EC430                                  COMPUTED  CENTER  OF  GRAVITY                          PAGE   2

 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
 1  0  0  0  0  0  0  0  0   0   0   0   0   0   0

        XG               YG               ZG               W
   0.0              0.0              0.0              0.900000E 07

 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
 1  1  1  1  1  1  1  1  1   1   0   0   0   0   0

        XG               YG               ZG               W
   0.176484E 01     0.0              0.101490E 00     0.106020E 08

 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
 1  1  1  1  0  0  1  1  0   1   0   0   1   1   1

        XG               YG               ZG               W
   0.196244E 01     0.0             -0.735070E-01     0.105160E 08

 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
 1  1  1  1  0  0  1  1  0   1   1   0   1   1   1

        XG               YG               ZG               W
   0.269753E 01     0.0             -0.728146E-01     0.106160E 08

 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
 1  1  1  1  0  0  1  1  0   1   0   1   1   1   1

        XG               YG               ZG               W
   0.119037E 01     0.0             -0.728146E-01     0.106160E 08

 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
 1  0  1  1  0  0  1  1  0   1   1   0   1   1   1

        XG               YG               ZG               W
   0.102122E 01     0.0             -0.358103E-01     0.104160E 08

 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
 1  0  0  1  0  0  1  1  0   1   1   0   1   1   1

        XG               YG               ZG               W
  -0.173353E 01     0.0              0.426318E-01     0.100160E 08

 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
 1  0  0  0  0  0  1  1  0   1   1   0   1   1   1

        XG               YG               ZG               W
   0.114180E 01     0.0              0.196114E 00     0.931600E 07
```

Figure 14. Output Variable Format, EC430 (cont.)

E. ZC 300, SUBMARINE THRUST

1. DESCRIPTION

This program solves equation 18 for axial thrust or force, $m\dot{u}$, and axial acceleration $\dot{u}$.

$$m\dot{u} = \frac{\rho}{2}\ell^2\left[a_i u^2 + b_i u u_c + c_i u_c^2\right] \qquad (18)$$

$\dot{u}$ = axial acceleration, (ft/sec^2)

$m\dot{u}$ = force, (lb)

m = mass, (slug $\frac{16\text{-sec}^2}{32.2\text{ ft}}$)

ρ = density, (1.995 slug/ft^3)

ℓ = length, (ft)

a_i, b_i, c_i = dimensionless constants depending on value of η, (u_c/u)

u = submarine axial velocity, (ft/sec)

u_c = command speed, (ft/sec)

Three sets of constants are used for a_i, b_i, and c_i depending on the value of η as compared to η high and η low in equation 19.

$$\begin{array}{ll} a_1\ b_1\ c_1 & \eta \geq \eta_{high} \\ a_2\ b_2\ c_2 & \eta_{low} \leq \eta \leq \eta_{high} \\ a_3\ b_3\ c_3 & \eta < \eta_{low} \end{array} \qquad (19)$$

At u = 0 equation 20 is used.

$$\begin{array}{ll} u_c \geq 0 & \eta = 1 \\ u_c < 0 & \eta = -1 \end{array} \qquad (20)$$

2. INPUT DATA DECK

The input variables are submarine length, mass, η high, η low, and the three values of a_i, b_i, and c_i. They are punched on cards according to table 11.

TABLE 11. INPUT DATA DECK, PROGRAM ZC300

Card	Column(s)	Format	Description
1	1-10	F10.3	AL, submarine length
	11-20	F10.3	AM, submarine mass
	21-30	F10.3	ETAHI, upper value of for coefficient change
	31-40	F10.3	ETALO, lower value of for coefficient change
2	1-10	F10.6	A_1
	11-20	F10.6	A_2
	21-30	F10.6	A_3
3	1-10	F10.6	B_1
	11-20	F10.6	B_2
	21-30	F10.6	B_3
4	1-10	F10.6	C_1
	11-20	F10.6	C_2
	21-30	F10.6	C_3

3. OUTPUT DATA

The output includes

a. All input variables

b. Table of force (thrust) as a function of command speed from -15 to +30 knots (Δ 5 knots) and for values of U from 0 to 30 knots (Δ 2.5 knots)

c. Table of acceleration as function of same velocity variables as b. above.

4. LISTINGS AND FLOW CHARTS

The listings and flow chart for program ZC300 are included in appendix I.

F. ZC690, ERROR CALCULATOR, DS + DR CONTROL

1. DESCRIPTION

This program calculates the percent error of change in a variable. Inputs are from EB920, Submarine Simulation Program - U at 60 seconds, theta high, phi mini, phi (peak near T = max), and psi at 60 seconds. The program calculates the change of each variable from value at t = 0 for the original program coefficients. Then a coefficient is set to zero and the changes recalculated and compared to the reference run as

$$\% \text{ change } (u) = 100\left[\frac{\Delta u_{ref} - \Delta u}{\Delta u_{ref}}\right]$$

These calculations are printed out with all input data for 5, 15, and 25 knots. Figure 15 shows the criteria used for comparison for each of the variables.

2. INPUT DATA DECK

Table 12 gives the format for the input data deck for this program.

TABLE 12. INPUT DATA DECK, PROGRAM ZC690

Card	Column(s)	Format	Description
1	1-4	I4	NO15, 1st run at 15 knots
	11-14	I4	NO25, 1st run at 25 knots
2	1-4	I4	Run number, NO
	11-18	2A4	Coefficient set to zero, COEF
	21-30	F10.5	Value of u at 60 sec, U60
	31-40	F10.6	Peak value of theta, THETHI
	41-50	F10.8	Minimum value of phi, PHIMIN
	51-60	F10.8	Peak value of last oscillation of phi, PHIUP
	61-70	F10.6	Value of psi at 60 sec., PSI60.
2-N	Repeat above card as desired		
N	Blank card for end of data		

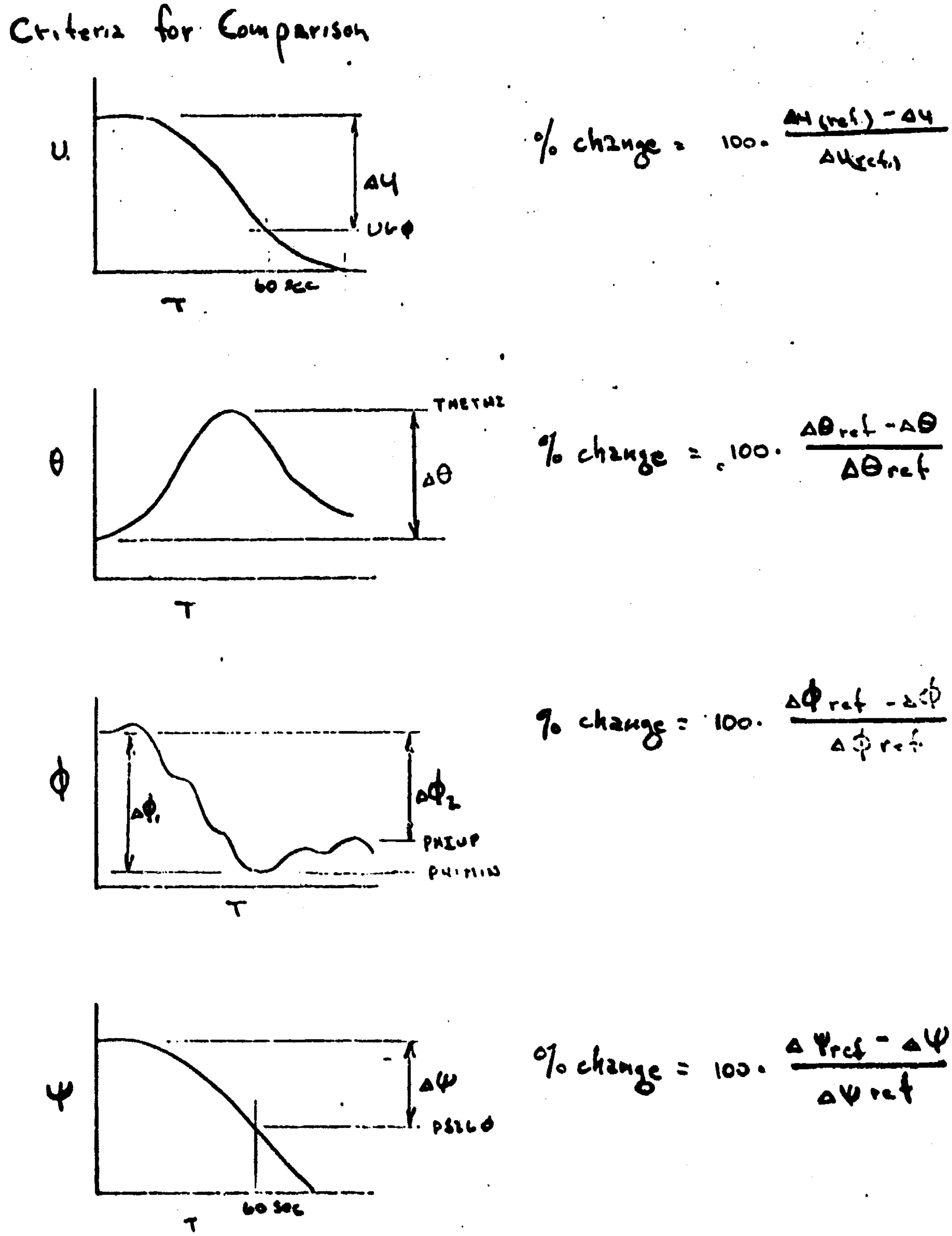


Figure 15. Comparison Criteria for Error Calculator

3. OUTPUT DATA

The output includes

a. Percent change in (see figure 15):

1. $\Delta u_{(60)}$

2. Δ theta$_{(peak)}$

3. Δ phi$_{(min)}$

4. Δ phi$_{(lobe)}$

5. Δ psi$_{(60)}$

b. All input variables

4. LISTINGS AND FLOW CHART

The listings and flow chart for program ZC690 are given in appendix A.

G. ZC691, ERROR CALCULATOR, DS CONTROL

1. DESCRIPTION

This program calculates the percent error or change in a variable compared to change of a reference. Inputs are from EB920, submarine simulation program. Reference data is data for u, Θ, and z from EB920 for original coefficients at 5, 15, and/or 25 knots. Comparable data, (see figure 15) is taken from the output of EB920 for setting a coefficient (COEF) to zero, or varying values of XG and ZG for center of gravity studies. The program calculates values such as:

$$\% \text{ change } \quad (u) \quad = \quad 100\left[\frac{\Delta u_{ref} - \Delta u}{\Delta u_{ref}}\right]$$

for u, Θ, and z

This program differs from ZC690 in that a new reference run can be used at each comparison rather than only one reference at the start. The reference run can be compared to either a single run (variable ISW2 = 1 in the program) or to a group of runs (ISW2 = 0).

2. INPUT DATA DECK

Table 13 gives the format of the input data deck for this program.

TABLE 13. INPUT DATA DECK, PROGRAM ZC691

Card	Column(s)	Format	Description
1	1-4	I4	Run number of first case at 15 knots (25 if no 15 knot case), NO15, right-adjust
	11-14	I4	Run number of first case at 25 knots, NO25, right adjust
	21	I1	ISW2, option for group or individual run comparison
2	1-4	I4	Run number, NO
	11-18	2A4	Coefficient set to zero (if used), COEF
	21-30	F10	Value of u min, U1
	31-40	F10	Value of theta max, THET1
	41-50	F10	Value of z at 60 sec., Z1

TABLE 13. INPUT DATA DECK, PROGRAM ZC691 (cont.)

Card	Column(s)	Format	Description
2	51-60	F10	Xg (if used), XG
	61-70	F10	Zg (if used), ZG
3-n	Repeat card No. 2 as desired for each comparison run		
n+1	New reference card for new (15 knot or 25 knot) speed range		
n-m	Repeat card No. 2 as desired for each comparison run		
m-k	Repeat cards n+1 and n-M for 25 knot speed range		
k+1	Blank card		

Table 14 and 15 give examples of the two types of input decks that can be used.

TABLE 14. DATA SUBMITTAL FOR ISW2 = 0, ZC691

Card	Use
1	Speed change run NO's. Comparison switch (ISW2 = 0)
2	Reference run data
3	Comparison run data
4-n	Repeat card 3 as required
n+1	New ref run data at next speed range
n+2	Comparison run data
n-m	Repeat card n+2 as required, etc.

TABLE 15. DATA SUBMITTAL FOR ISW2 = 1, EC961

Card	Use
1	Speed change run numbers. Comparison switch (ISW2 = 1)
2	Reference run data
3	Comparison run data
4	Reference run data
5	Comparison run data
6-n	Repeat card 2 as required right through speed changes
7-n+1	Repeat card 3 as required right through speed changes

3. OUTPUT DATA

The output includes

a. Percent change in

1. Δ u

2. Δ theta

3. Δ z

b. All input quantities and initial conditions

4. LISTINGS AND FLOW CHART

The listings and flow chart for program [illegible] are given in Appendix A.

H. EC140, ROOT CRACKER PROGRAM, LONGITUDINAL

1. DESCRIPTION

This program is used to help analyze submarine elevator impulse response for natural frequencies and damping factors. The program solves the roots of the characteristic equation which in turn was solved from a matrix of the longitudinal force terms of the equations of motion.

These roots are equivalent to the values α, β, and γ of equation (21).

$$y = a_1 e^{-\gamma t} + a_2 e^{-\alpha t} \cos(\beta t + \psi) \tag{21}$$

which is the longitudinal time output response to an impulse input. The variables a_1, a_2, and ψ (as well as an additional estimate of α, β, and γ) can be estimated from program EC330, Time Response Coefficient Estimator, Longitudinal. Then the frequency response to a particular impulse as described by equation (21) can be calculated and plotted in program EC310, Brown's Convergence and Comparative Plot Program, against an actual impulse plot made from an EB920, Submarine Simulation Program run.

Equation (22) shows the matrix formed from the longitudinal terms of the equations of motion. The assumptions and modifications that were used to simplify the matrix are noted. The values of $\bar{w}$ and $\bar{q}$ are average values of these terms as taken from a number of actual full set runs of a submarine at steady state conditions on program EB920, Submarine Simulation Program. These values are used as a compensation for the non-linear terms in the longitudinal loop by modifying the basic coefficients. For example,

$$Z_{w|q|}\, w\bar{q} \xrightarrow{\text{replaces}} Z_{w|q|}\, w|q|$$

where $\bar{q}$ is an average value of $|q|$ used as a constant.

Several of the coefficients shown in the matrix of equation (22) are combined inside this program to provide more simple matrix elements. These are used in the actual matrix solved and are printed out for convenience. These new values of combined coefficients will be shown as primes here (are printed without primes) such as

$$M_w' = M_w + \frac{M_{w|w|}\,\bar{w}}{u}$$

Matrix in S, Longitudinal Case, (22)

$$\begin{vmatrix} s^2\left(\frac{\rho}{2}\ell^5 M_{\dot q} - I_y\right) + s\left(\frac{\rho}{2}\ell^4 M_q u_0 + \frac{\rho}{2}\ell^4 M_{|w|q}\bar{\omega}\right) + z_B B & s^2\left(\frac{\rho}{2}\ell^4 M_{\dot w}\right) + s\left(\frac{\rho}{2}\ell^3 M_w u_0 + \frac{\rho}{2}\ell^3 M_{w|w|}\bar{\omega}\right) & s\left(\frac{\rho}{2}\ell^3 2 M_{*} u_0 + \frac{\rho}{2}\ell^3 M_{*|w|}\bar{\omega}\right) \\ s\left(\frac{\rho}{2}\ell^4 Z_{\dot q}\right) + \left(m u_0 + \frac{\rho}{2}\ell^3 Z_q u_0\right) & s\left(\frac{\rho}{2}\ell^3 Z_{\dot w} - m\right) + \left(\frac{\rho}{2}\ell^3 Z_{w|q|}\bar{q} + \frac{\rho}{2}\ell^2 Z_w u_0 + \frac{\rho}{2}\ell^2 Z_{w|w|}\bar{\omega}\right) & \frac{\rho}{2}\ell^2 2 Z_{*} u_0 \\ \frac{\rho}{2}\ell^4 X_{qq}\bar{q} + \frac{\rho}{2}\ell^3 X_{wq}\bar{\omega} - m\bar{\omega} & -0- & s\left(\frac{\rho}{2}\ell^3 X_{\dot u} - m\right) + \left(\frac{\rho}{2}\ell^2 2 a_i u_0 + \frac{\rho}{2}\ell^2 b_i u_0\right) \end{vmatrix} \begin{vmatrix} q_{\text{Pitch}} \\ w_{\text{Normal}} \\ u_{\text{Axial}} \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \\ 0 \end{vmatrix}$$

From "Standard Equations of Motion for Submarine Simulation"

$X_B = X_G = Y_G = Z_G = 0$	$uq = u_0 q$	$X_{uu} = 0$
$r = \phi = p = 0$	$u^2 = u_0^2$	$Z_{ww} = 0$
$W = B$	$uw = u_0 w$	$M_{q|q|} = 0$
$\delta_r = \delta_s = \delta_b = 0$	$|w w| = \bar{w}\, w$	$M_{ww} = 0$
$I_{xy} = I_{zx} = I_{yz} = 0$		$Z_{|w|} = 0$
All η-terms $= 0$		$M_{|w|} = 0$

The matrix element, (1, 2), equation (22), is then carried to the matrix as

$$s^2\left(\frac{\rho}{2} l^4 M_{\dot{w}}\right) + s\left(\frac{\rho}{2} l^3 M_w'\right)$$

This same simplification is carried out for the other modified coefficients

$$M_q' = M_q + \frac{M_{|w|q}\,\bar{w}}{u}$$

and

$$Z_w' = Z_w + \frac{l\,Z_{w|q|}\,\bar{q}}{u} + \frac{Z_{w|w|}\,\bar{w}}{u}$$

The characteristic equation resulting from the expansion of equation (22) is a fourth-order equation which has four roots,

$$\alpha + j\beta, \quad \alpha - j\beta \qquad \gamma + j0, \quad \delta + j0$$

α and β are the real and imaginary components of the complex roots. γ is the larger of the two real roots, δ is the smaller. Only three roots are used in the analysis of longitudinal natural frequencies and damping factor. The fourth root (smallest real magnitude) has a very small coefficient in the time domain and represents a long term effect due to changes in U (forward speed). This term can thus be ignored when considering pitch motions, but the other roots are more accurate if this term is included in the matrix.

Certain measurement parameters are calculated from the roots by this program. They are the natural frequency

$$\omega_n = \sqrt{\alpha^2 + \beta^2}$$

the time to damp to one-half

$$T_{1/2} = \frac{\ln 2}{\alpha}$$

the period

$$P = \frac{2\pi}{\beta}$$

and the damping ratio

$$\zeta = \frac{\alpha}{\omega_n} = \frac{\alpha}{\sqrt{\alpha^2 + \beta^2}}$$

2. SUBROUTINE DESCRIPTIONS

a. PLACE - This subroutine places each set of elements of the matrix into the matrix so that the characteristic equation can be expanded.

b. CHREQN - This subroutine expands the matrix formed by subroutine PLACE into the characteristic equation by taking the determinant of the matrix.

c. MULLER - This subroutine, by the method of Muller, calculates the roots of the characteristic equation.

3. INPUT DATA DECK

The input variables to the program include the coefficients and submarine constants noted in the matrix of equation (22). They are identified in detail with their locations in table 16.

TABLE 16. INPUT DATA DECK, PROGRAM EC140

Card	Column(s)	Format	Description
1	1-80	8F10.5	AWD, ZW, ZQD, ZQ, MWD, MW, MQD, MQ
2	1-10	F10.5	AW, $\bar{w}$, average value of normal component of submarine velocity, based on a number of full set runs, feet/second
2	11-20	F10.5	AQ, $\bar{q}$, average pitch rate, radians/second
2	21-60	4F10.5	AQ, $\bar{q}$, average pitch rate, radians/second
3	1-10	F10.5	XUD
3	11-20	F10.5	A11, a_i, thrust coefficient
3	21-30	F10.5	A12, b_i, thrust coefficient
3	31-70	4F10.5	ZSTR, MSTR, XQQ, XWQ
4	1-10	F10.5	L, submarine length, feet

TABLE 16. INPUT DATA DECK, PROGRAM EC140 (cont.)

Card	Column(s)	Format	Description
4	11-20	F10.5	M, submarine mass, slugs
4	21-30	F10.5	IY, moment of inertia about the y-axis, slug-ft^2
4	31-40	F10.5	B, submarine buoyancy, pounds
4	41-50	F10.5	ZB, z_B, z-component of center of buoyancy, feet
4	51-60	F10.5	RHO, ρ, density of seawater, slugs/ft^3
5	1-10	F10.5	U, u, forward velocity, knots Stack as many U cards as desired. Place a blank card after the last U card for that set. Repeat all above cards as desired for a new case with different coefficients.
last	1-4	F4.3	999. The program takes normal end of job with this card.

4. OUTPUT DATA

Figure 16 shows a typical output sheet from this program.

a. Input data - all input data is printed out in the units described above.

b. The factors A1, A2, etc., of the characteristic equation are printed out. For this 4th order equation

$$A1\, s^4 + A2\, s^3 + A3\, s^2 + A4\, s + A5 = 0$$

c. Roots - The four roots of the characteristic equation are printed out.

d. Measurement Parameters - The natural frequency, time to damp to one-half, period and the damping ratio are printed out.

5. LISTINGS AND FLOW CHART

The listings and flow chart for program EC140 are given in Appendix A.

EC140						PAGE	1
ZWC	ZW	ZQD	ZQ	MWD	MW	MQD	MQ
-0.140000E-01	-0.170000E-01	-0.100000E-03	-0.700000E-02	-0.130000E-03	0.750000E-02	-0.700000E-03	-0.400000E-0[illegible]
AW	AQ	MWAQ	MWAW	ZWAQ	ZWAW		
0.200000E-01	0.0	-0.200000E-02	-0.140000E-01	-0.100000E-01	-0.550000E-01		
XUC	A11	A12	ZSTR	MSTR	XQQ	XWQ	
-0.400000E-03	-0.150000E-0[illegible]	-0.130000E-02	0.220000E-03	0.750000E-04	-0.100000E-03	-0.140000E-01	
I	M	IY	R	ZR	WHO		
0.175000E 01	0.300000E 0[illegible]	0.130000E 1[illegible]	0.110000E 08	-0.160000E 01	0.200000E 01		
MW	MQ	Z[illegible]					
A1	A2	A3	A4	A5	[illegible]		
[illegible]	I	[illegible]	I	[illegible]	I	2	I
[illegible]	T(1/2)	2	0				
0.7466846E-02	-0.430710E-0[illegible]	-0.171300E-01					
-0.4374740 01	-0.4773920 0[illegible]	-0.3432310 1[illegible]	-0.8226410 17	-0.4307030 15	0.944000E 01		
-0.3571950-02	-0.1113660-1[illegible]	-0.3772420-01	-0.6345630-01	-0.3772400-01	0.6345630-01	-0.2524370-01	0.7054750-01
0.712405E-01	0.183703E 0[illegible]	0.113020E 03	0.522382E 00				
0.7483426E-02	-0.430355E-0[illegible]	-0.1726516-01					
-0.4374740 01	-0.9547660 0[illegible]	-0.4362440 1[illegible]	-0.1729390 18	-0.1315520 16	0.168000E 02		
-0.1959270-01	-0.3507360-01	-0.1359370-01	0.3517360-01	-0.1470970-01	0.3392850-0[illegible]	-0.162350 0[illegible]	-0.1472470-0[illegible]
0.401750E-01	0.352703E 0[illegible]	0.179143E 03	0.497634E 00				
0.7489946E-02	-0.430336E-0[illegible]	-0.171534E-01					
-0.4374740 01	-0.1531530 01	-0.7596621 1[illegible]	-0.2746910 18	-0.4394430 16	0.253350E 02		
-0.1508740-01	-0.3511150-01	-0.1535740-01	0.3511150-01	-0.2474250-01	0.2102760-0[illegible]	-0.272330 0[illegible]	-0.4557360-0[illegible]
0.182138E-01	0.459475E 0[illegible]	0.174043E 03	0.334685E 00				

Figure 16. Output Variable Format, EC140

I. EC320 ROOT CRACKER PROGRAM, LATERAL

1. DESCRIPTION

This program is used to help analyze submarine rudder impulse response for natural frequencies and damping factors. The program solves for the roots of the characteristics equation which in turn was solved from a matrix of the lateral force terms of the equations of motion.

These roots are equivalent to the values α, β, γ, and δ of the equation below:

$$y = a_1 e^{-\gamma t} + a_2 e^{-\alpha t} \cos(\beta t + \psi) + a_3 e^{-\delta t}$$

which is the lateral time output response to an impulse input. The variables a_1, a_2, a_3, and ψ can be estimated from program EC150, Time Response Coefficient Estimator, Lateral. Then the frequency response to a particular impulse, can be calculated and plotted in program EC310, Browns Convergency and Comparative Plot Program, against an actual impulse plot made from an EB920, Submarine Simulation Program run.

Equation (23) shows the matrix formed from the lateral terms of the equations of motion. The assumptions and modifications that were used to simplify the matrix are noted. The values of $\bar{p}$, $\bar{r}$, and $\bar{v}$ are average values of these terms as taken from a number of actual full set runs of the submarine at steady state conditions on program EB920, Submarine Simulation Program. Figure 17 qualitatively shows the errors inherent in the substitution of one of the values, say $\bar{r}$ for $|r|$. These values are used as a compensation for the non-linear terms in the lateral loop by modifying the basic coefficients. For example,

$$Y_{v|v|} v \cdot \bar{v} \xrightarrow{\text{replaces}} Y_{v|v|} v \left| (v^2 + w^2)^{1/2} \right|$$

where $\bar{v}$ is an average value of $|v|$ used as a constant.

Y_v and $Y_{v|v|}$ are combined into a single term in the matrix. A similar process is used for $|p|$ and $|r|$, set to $\bar{p}$ and $\bar{r}$ respectively.

Several of the coefficients shown in the matrix are combined within this program to provide more simple matrix elements. These are used in the actual matrix solved and are printed out for convenience. These new values of combined coefficients will be shown as primes here (are printed without primes) such as:

$$Y_v' = Y_v + \frac{Y_{v|v|} \bar{v} + \ell Y_{v|r|} \bar{r}}{u}$$

Matrix in S, Lateral Case (23)

$$
\begin{vmatrix}
s\left(\frac{\rho}{2}\ell^3 Y_{\dot v}-m\right)+\frac{\rho}{2}\ell^3 Y_{v|r|}\bar r + \frac{\rho}{2}\ell^2 Y_v u_0 + \frac{\rho}{2}\ell^2 Y_{v|v|}\bar v & s\left(\frac{\rho}{2}\ell^4 Y_{\dot p}\right)+\frac{\rho}{2}\ell^3 Y_p u_0 & s\left(\frac{\rho}{2}\ell^4 Y_{\dot r}\right) - m u_0 + \frac{\rho}{2}\ell^3 Y_r u_0 \\
s^2\left(\frac{\rho}{2}\ell^4 K_{\dot v}\right) + s\left[\frac{\rho}{2}\ell^3\left(K_v u_0 + K_{v|v|}\bar v\right)\right] & s^2\left(\frac{\rho}{2}\ell^5 K_{\dot p}-I_x\right)+z_B B + s\left(\frac{\rho}{2}\ell^5 K_{p|p|}\bar p+\frac{\rho}{2}\ell^4 K_p u_0\right) & s^2\left(\frac{\rho}{2}\ell^5 K_{\dot r}\right)+s\left(\frac{\rho}{2}\ell^4 K_r u_0\right) \\
s\left(\frac{\rho}{2}\ell^4 N_{\dot v}\right)+\frac{\rho}{2}\ell^3 N_v u_0 + \frac{\rho}{2}\ell^3 N_{v|v|}\bar v & s\left(\frac{\rho}{2}\ell^5 N_{\dot p}+\frac{\rho}{2}\ell^4 N_p u_0\right) & s\left(\frac{\rho}{2}\ell^5 N_{\dot r}-I_z\right)+\frac{\rho}{2}\ell^4\left(\ell N_{r|r|}\bar r + N_r u_0 + N_{v|r|}\bar v\right)
\end{vmatrix}
\begin{vmatrix}
\underset{\text{Lateral}}{v} \\ \underset{\text{Roll}}{p} \\ \underset{\text{Yaw}}{r}
\end{vmatrix}
=
\begin{vmatrix} 0 \\ 0 \\ 0 \end{vmatrix}
$$

From "Standard Equations of Motion for Submarine Simulation"

$x_B = x_G = Y_G = Z_G = Y_B = 0$

$q = w = 0$

$W = B$

$\delta_r = \delta_s = \delta_B = 0$

All η-terms $= 0$

$Y_{p|p|} = Y_* = K_* = N_{r|r|} = 0$

$ur = u_0 r$	$vr = r_0 v$
$up = u_0 p$	$pr = r_0 p$
$uv = u_0 v$	$ur = u_0 r$
$uu = u_0 u$	$rr = r_0 r$

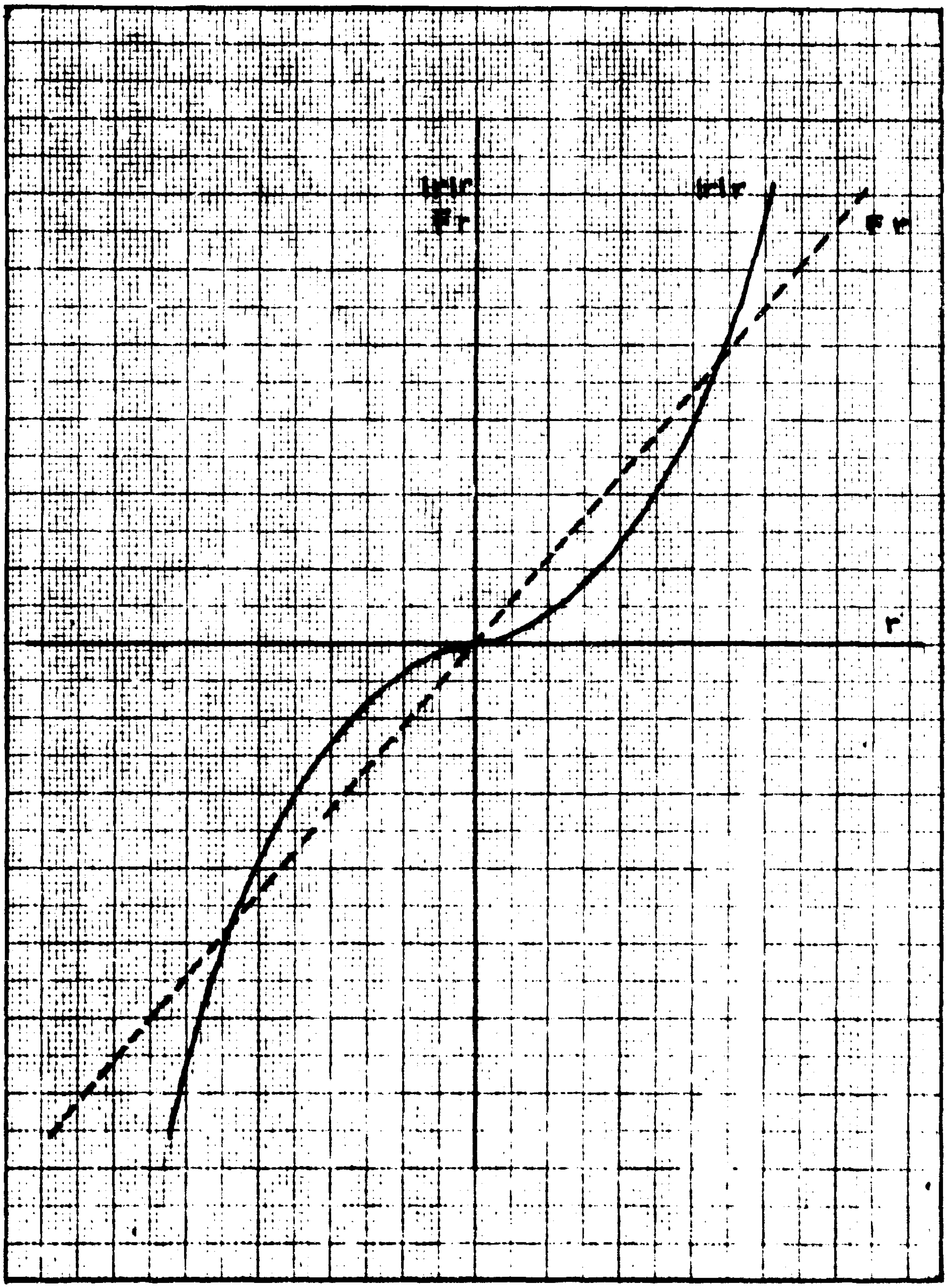


Figure 17. Error in Use of $\bar{r}$

The matrix element, (1,1) equation (1), is then carried to the matrix as

$$s\left(\frac{\rho}{2} l^3 Y_{\dot{v}} - m\right) + \frac{\rho}{2} l^3 u Y_v'$$

This same simplification is carried out for the other modified coefficients.

$$Y_p' = Y_p + \frac{l\, Y_{p|p|}\, \bar{p}}{u}$$

$$K_v' = K_v + \frac{K_{v|v|}\, \bar{v}}{u}$$

$$K_p' = K_p + \frac{l\, K_{p|p|}\, \bar{p}}{}$$

$$N_v' = N_v + \frac{N_{v|v|}\, \bar{v}}{u}$$

$$N_r' = N_r + \frac{l N_{r|r|}\, \bar{r} + N_{r|v|}\, \bar{v}}{u}$$

The characteristic equation resulting from the expansion of equation (23) is a fourth-order equation which has four roots

$$\alpha + j\beta \quad \alpha - j\beta \quad \gamma + j0 \quad \delta + j0$$

which can be used in impulse response equation.

2. SUBROUTINE DESCRIPTIONS

a. PLACE - This routine places each element of the matrix similar to table in its location in the matrix ready to expand into the characteristic equation.

b. CHREQN - This subroutine expands the matrix formed by subroutine PLACE into the characteristic equation by taking the determinant of the matrix.

c. MULLER - This subroutine, by the method of Muller, calculates the roots of the characteristic equation.

3. INPUT DATA DECK

The input variables to the program include the coefficient and submarine constants noted in the matrix of equation (23). They are identified in detail with their locations in table 17.

TABLE 17. INPUT DATA DECK, PROGRAM EC320

Card	Column(s)	Format	Description
1	1-60	6F10.5	YVD, YV, YPD, YP, YRD, YR
2	1-60	6F10.5	AKVD, AKV, AKPD, AKP, AKRD, AKR
3.	1-60	6F10.5	ANVD, ANV, ANPD, ANP, ANRD, ANR
4	1-10	F10.5	AL, submarine length, feet
4	11-20	F10.5	AM, submarine mass, slugs
4	21-30	F10.5	IX, movement of inertia about the X-axis, slug-ft^2
4	31-40	F10.5	IZ, movement of inertia about the axis slug-ft^2
4	41-50	F10.5	B, buoyancy, pounds
4	51-60	F10.5	ZB, z-component of center of buoyancy - feet
4	61-70	F10.5	RHO, ρ density of sea water-slugs/ft^3
5	1-10	8F10.5	YVAV, YVAR, YPAP, AKVAV, AKPAP, ANVAV, ANRAR, ANAVR
6	1-10	F10.5	RBAR, $\bar{r}$, average yaw rate, based on a number of full set runs, radians/second
6	11-20	F10.5	VBAR, $\bar{v}$, average normal velocity, feet/second
6	21-30	F10.5	PBAR, $\bar{p}$, average roll rate, radians/second
7	1-5	I5	IDIV, this option allows the program user to divide YVAV*VBAR, YVAR*RBAR, YPAP*PBAR, KVAV*VBAR, KPAP*PBAR, NVAV*VBAR, NAVR*RBAR and NAVR*VBAR by velocity, before punching cards or letting the computer do the division.

TABLE 17. INPUT DATA DECK, PROGRAM EC320 (cont.)

Card	Column(s)	Format	Description
			0 User to divide 1 Computer divides by
8	1-10	F10.5	forward velocity, knots. Stack as many per set of coefficients as desired. Place a blank card after last card of each set. This causes next case to be read.
last	1-5	F5.4	9999. The program takes normal end of job with this card.

4. OUTPUT DATA

Figure 18 of program EC140 shows a typical output from this program except that the measurement parameters are not calculated.

a. Input data - all input data (except IDIV) is printed out in the units detailed above.

b. The factors A1, A2, etc., of the characteristics equation are printed out. For this 4th-order equation

$$A1\,s^4 + A2\,s^3 + A3\,s^2 + A4\,s + A5 = 0$$

c. Roots - the four roots of the characteristic equation are printed out.

5. LISTINGS AND FLOW CHART

The listings and flow chart for the main program of EC320 are given in appendix A. The subroutine listing and flow charts are identical to those of program EC140.

```
                                         EC320                                                PAGE      1

     YV?            VV            VPM            VP            YPD            YP
-1.150000E-01  -0.200000E-01  -0.750000E-04  -0.700000E-03  0.100000E-03  0.400000E-12
     KVP            KV            KPD            KP            KPD            KP
-0.960000E-04  -0.200000E-03  -0.200000E-04  -0.700000E-04  -0.130000E-04  0.0
     NVP            NV            NPD            NP            NPD            NP
 0.200 [illegible] -05  -0.100000E-01  -0.200000E-04  -0.200000E-03  -0.750000E-03  -0.400000E-12
      L              M             IX             IZ             R             ZP            RHO
 0.275000 [illegible] 03  0.300000E 04  0.360000E [illegible]  0.130000E 10  0.100000E 02  -0.14 [illegible] 01  0.200000E 01
    YVAV           YVAP           YPAP           KVAV           KPAP           NVAV           NPAP           NAVR
-0.750000E-01  -0.100000E-01  0.0            -0.230000E-02  -0.200000E-05  0.120000E-01  0.0            -0.33 [illegible]
    RBAR           YPAR           PBAR
 0.250000E-01   0.200000E 01   0.100000E-02

     YV             YP             KV             KP             NV             NP
     A1             A2             A3             A4             A5             U
     R              I              P              I              P              I

-1.329515E-01  -0.700000E-03  -0.472351E-03  -0.713125E-04  -0.257903E-02  -0.420076E-12
-0.101273E 24  -0.142783E 23  -0.170789E 23  -0.542135E 22  -0.122057E 21  0.1 [illegible] 12
-0.203733E-01   0.0           -0.220377E [illegible]  -0.733561E-14  -0.473666E-01  0.48 [illegible] 01  -0.474666E-01  -0.427725E 00

-1.296347E-01  -0.700000E-03  -0.381557E-03  -0.700717E-04  -0.225269E-02  -0.42 [illegible] 12
-0.101273E 24  -0.45 [illegible] 23  -0.208012E 23  -0.232257E 22  -0.122691E 21  0.251352E 12
-1.155747E-01   0.0           -0.223 [illegible]  -0.444555E 00  -0.723094E-01  0.484655E 00  -0.323 [illegible]  0.47 [illegible]-24

-1.264757E-01  -0.700000E-03  -0.336175E-03  -0.7221 [illegible]-04  -0.228652E-02  -0.41 [illegible] 12
-0.101274E 24  -0.4 [illegible] 23  -0.334 [illegible] 23  -0.127489E 23  -0.112366E 21  0.327 [illegible] 12
-0.108 [illegible]-01   0.0           -0.270 [illegible]-01  -0.493650E 00  -0.373293E-01  0.48 [illegible]  -0.42 [illegible]  0.415244E-25
```

Figure 18. Output Variable Format, EC320

J. EC150, COEFFICIENT ESTIMATOR, LATERAL

1. DESCRIPTION

The response to a rudder impulse (lateral case) of a submarine can be approximated by the equation (24)

$$y(t) = a_1 e^{-\gamma t} + a_2 e^{-\alpha t} \cos(\beta t + \psi) + a_3 e^{-\delta t} \tag{24}$$

where
- y = roll angle of the submarine, radians
- t = time, seconds
- γ, α, δ = roots of the characteristic equation or damping factor of the individual components
- β = frequency of oscillating term, radians/second
- ψ = phase delay, radians
- a_1, a_2, a_3 = coefficients of magnitude of each term

It is often quite useful to determine the magnitude of all the terms on the right side of equation (24) for a particular response to a lateral control impulse to a submarine in motion.

Values for y(t) can be calculated by solving the detailed "Standard Equations of Motion for Submarine Simulation". GAC program EB920 will perform these calculations over the time period desired. This program will also punch output cards with time and ϕ . This data deck will be used later. Program EB920 further provides a computer plot of ϕ versus time as shown in figure 19 when CALCOMP plotter subroutines are available.

Program EC320, Root Cracker Program (Lateral Case), solves for the roots of the characteristic equation using the matrix Laplace form taken from the equations of motion. These roots are used for initial values α , β , γ , and δ in equation (24) above. They are not exact because of the program linearization. These values are inputs to program EC150, time response coefficient estimator, lateral, which will estimate the values a_1 , a_2 , a_3 , and ψ . The punched data deck from EB920 is the only other input data required.

The program calculates the following:

a. To find ψ :

At t_4 (figure 19), the actual detailed calculated curve has decayed sufficiently that, for this calculation, the a_1 and a_3 terms may be considered zero. For all response curves having the first lobe after t = 0, the a_2 term is negative. Therefore, the positive peaks of figure 19 are equivalent to peak negative (-) values of the cosine term

$$\cos(\beta t_4 - \psi) = -1 \;;\; \beta t_4 - \psi = \pi \tag{25}$$

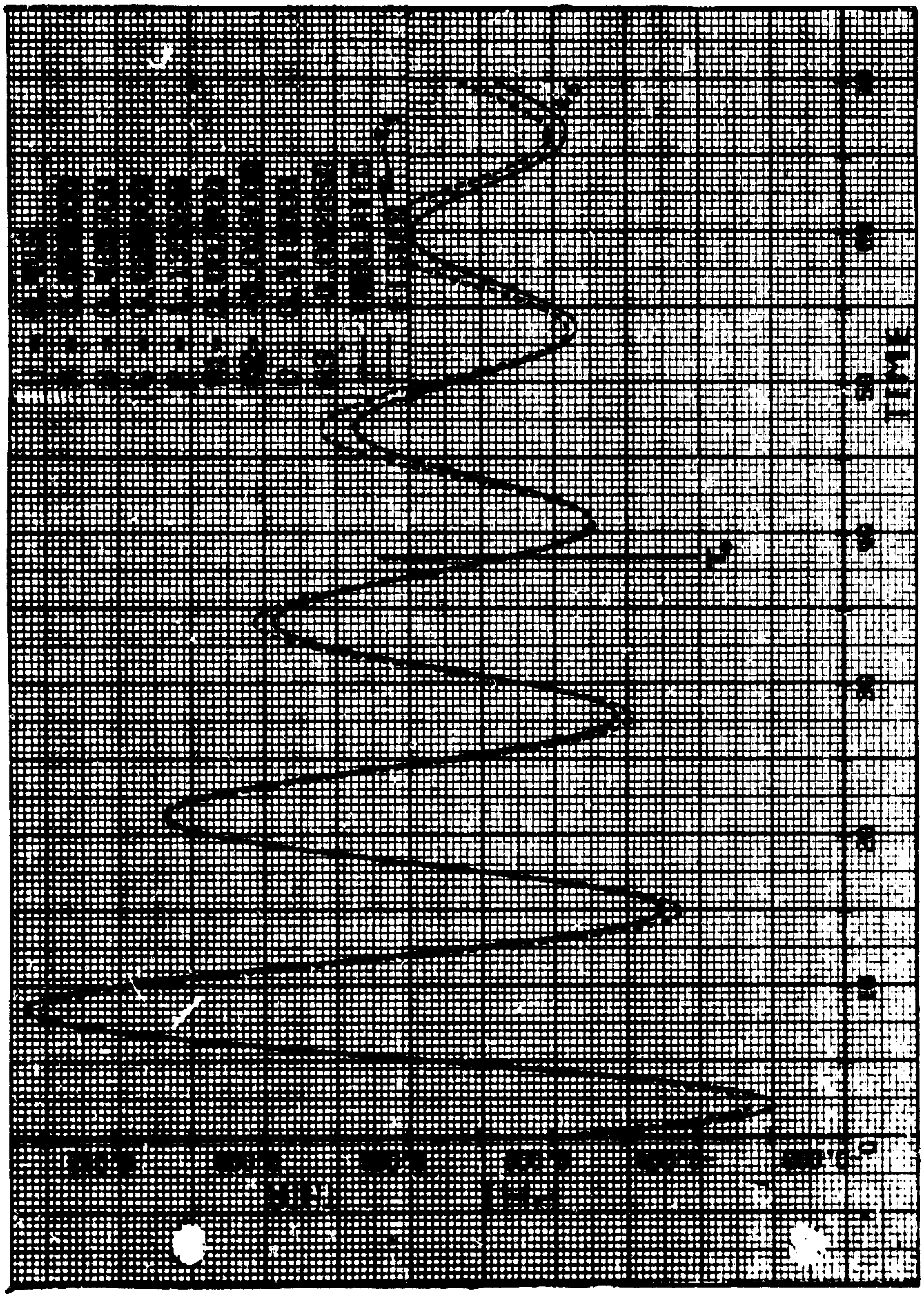


Figure 19. Submarine Response Curve, Lateral Case

or

$$\psi = (\beta t_4 - \pi) - 2\pi n \; ; \; n = 0, 1, \cdots \qquad (26)$$

The last term of equation (26) merely subtracts multiples of 360° from this calculated value of ψ to keep it an equivalent smallest number. This program automatically uses the value of the peak of the last lobe (searched out from the data deck) for the magnitude and time of t_4.

b. Time, T_o, at which the a_3 term of equation (24) is negligible:

$$T_o = \frac{4.61}{\delta} \qquad (27)$$

The printout data will provide T_o and also the values of t_4 and t_5. Value of T_o should be checked to verify that it is less than t_4. Otherwise, EB920 may have to be run for a longer time to provide good input data for this program.

c. To find a_2:

Using values of t_4 and t_5 (selected automatically from the data deck input by this program) and assuming that the a_3 term is zero:

$$y_4 = a_1 e^{-\gamma t_4} - a_2 e^{-\alpha t_4} \qquad (28)$$

$$y_5 = a_1 e^{-\gamma t_5} - a_2 e^{-\alpha t_5} \qquad (29)$$

$$a_2 = -\frac{y_4 - y_5 e^{-\gamma(t_5 - t_4)}}{e^{\gamma(t_5 - t_4) - \alpha t_5} + e^{-\alpha t_4}} \qquad (30)$$

d. To find a_1:

Substitute the value of a_2 just found into equation (27)

$$a_1 = \frac{y_4 + a_2 e^{-\alpha t_4}}{e^{-\alpha t_4}} \qquad (31)$$

e. To find a_3:

At $t = 0$, equation (24) can be arranged as:

$$a_3 = -(a_1 + a_2 \cos\psi) \qquad (32)$$

These values ψ, a_1, a_2, and a_3 are outputs of the program EC150.

They may then be used, with the values of α, β, γ, and δ from the Root Cracker program in EC310, Curve Fitting Program. This program will take these eight terms, calculate a set of values versus time, and plot against the exact calculated response values of EB920 for comparison as shown in figure 19.

Program EC310 has an option for convergence of six of the terms of equation (24) for optimum values by use of Brown's routine. δ and a_3 are usually held constant. This option can be exercised separately and plots run. The function will sometimes converge more closely than figure 19, and a very satisfactory set of values for the terms of equation (24) will be available. However, due to the approximations involved in EC320 and EC150, the function (which is converged through Brown's routine and needs very close initial values of all terms) may not converge. In this event, additional calculations or intuitive estimates for closer values of some or all terms, α, β, γ, δ, ψ, a_1, and a_3 may need to be made and the last one or two steps above repeated.

Even though final convergence may not be achieved with terms calculated through EC320 and EC150, the first plot of estimated values (of the nature of figure 19) is a very useful starting place for final convergence attempts.

2. INPUT DATA DECK

The input data consists principally of:

a. Punched cards from impulse run of EB920 Submarine Simulation Program punched at two second intervals.

b. Values for α, β, γ, and δ from EC320, Root Cracker, 3D, Lateral program.

The data deck format is given in table 18.

TABLE 18. INPUT DATA DECK, PROGRAM EC150

Card	Column(s)	Format	Description
1	1-5	I5	N, number of data points from EB920 run (one per punched card)
2	1-10	F10.5	U, submarine speed, knots
2	11-18	2A4	NAME, use the word, PHI. For use as y-axis label on tabulated printout data sheets
3-n	1-10	E15.7	PHI(I) values on punched cards from EB920, radians

TABLE 18. INPUT DATA DECK, PROGRAM EC150 (cont.)

Card	Column(s)	Format	Description
3-n	11-20	E15.7	T values on punched cards from EB920, seconds
n+1	1-10	F10.5	A, α, value of root calculated in EC320. Input value must be a positive number
n+1	11-20	F10.5	B, β, value of root from EC320, input as a positive number, radians/second
n+1	21-30	F10.5	G, γ, value of root from EC320, input as a positive number
n+1	31-40	F10.5	D, δ, value of root from EC320, input as a positive number
n+2			Blank card for normal end of job

4. OUTPUT DATA

The output includes sequentially numbered papers each identified with EC150. The following terms are printed:

U - submarine speed, knots, PHI(I) - all the values of the impulse run of EB920 as read by the card inputs to this program

YMIN - the value of PHI at t_5 (see figure 19) as selected from the input cards from EB920, radians

YMAX - the value of PHI at t_4, radians

TO - T_0, time at which the a_3 term of equation (24) is negligible, seconds

T4 - t_4, last peak value of oscillating function as selected from the input cards from EB920, seconds

T5 - t_5, last minimum value of oscillating function, seconds

A - α, value from EC320

B - β, value from EC320

G - γ, value from EC320

P - Ψ or PSI, calculated phase delay, radians

A1 - a_1, calculated constant for equation (24)

A2 - a_2, calculated constant for equation (24)

D - δ, value from EC320

A3 - a_3, calculated constant for equation (24)

5. LISTINGS AND FLOW CHART

The listings and flow chart for program EC150 are given in appendix A.

K. EC330, COEFFICIENT ESTIMATOR, LONGITUDINAL

1. DESCRIPTION

The response to an elevator impulse (longitudinal case) of a submarine can be approximated by equation (33)

$$y(t) = a_1 e^{-\gamma t} + a_2 e^{-\alpha t} \cos(\beta t + \psi) \quad (33)$$

where

y = pitch angle (theta) of the submarine, radians

t = time, seconds

α, γ = roots of the characteristic equation or damping factor of the individual components

β = frequency of oscillating term, radians/second

ψ = phase delay, radians

a_1, a_2 = dimensionless coefficients of magnitude for each term

It is sometimes necessary to determine the magnitude of all the terms on the right side of equation (33) for a particular response to a longitudinal control impulse to a submarine in motion. This program provides a method for securing first, reasonable approximations of the terms:

$$\alpha, \beta, \gamma, \psi, a_1 \text{ and } a_2$$

The exact response to a specific impulse can be calculated by solving the equations of motion. GAC program EB920 will perform these calculations over the time period desired. This program (EB920), in addition, will punch cards of the output value, Y or theta, versus time. (Time intervals of punch data should be two seconds for use in EC330). This data deck will be used as an input to the subject program, EC330. Program EB920 further can provide a computer plot of Y (THETA) versus time as shown in figure 20.

The computer makes the following sets of calculations:

β - The first approximation for (See figure 20) is taken as a half cycle for the period t_1 through t_2. The computer searches the input data for these two points and takes the differences

$$P = t_2 - t_1 \quad ; \quad \beta = \pi / P$$

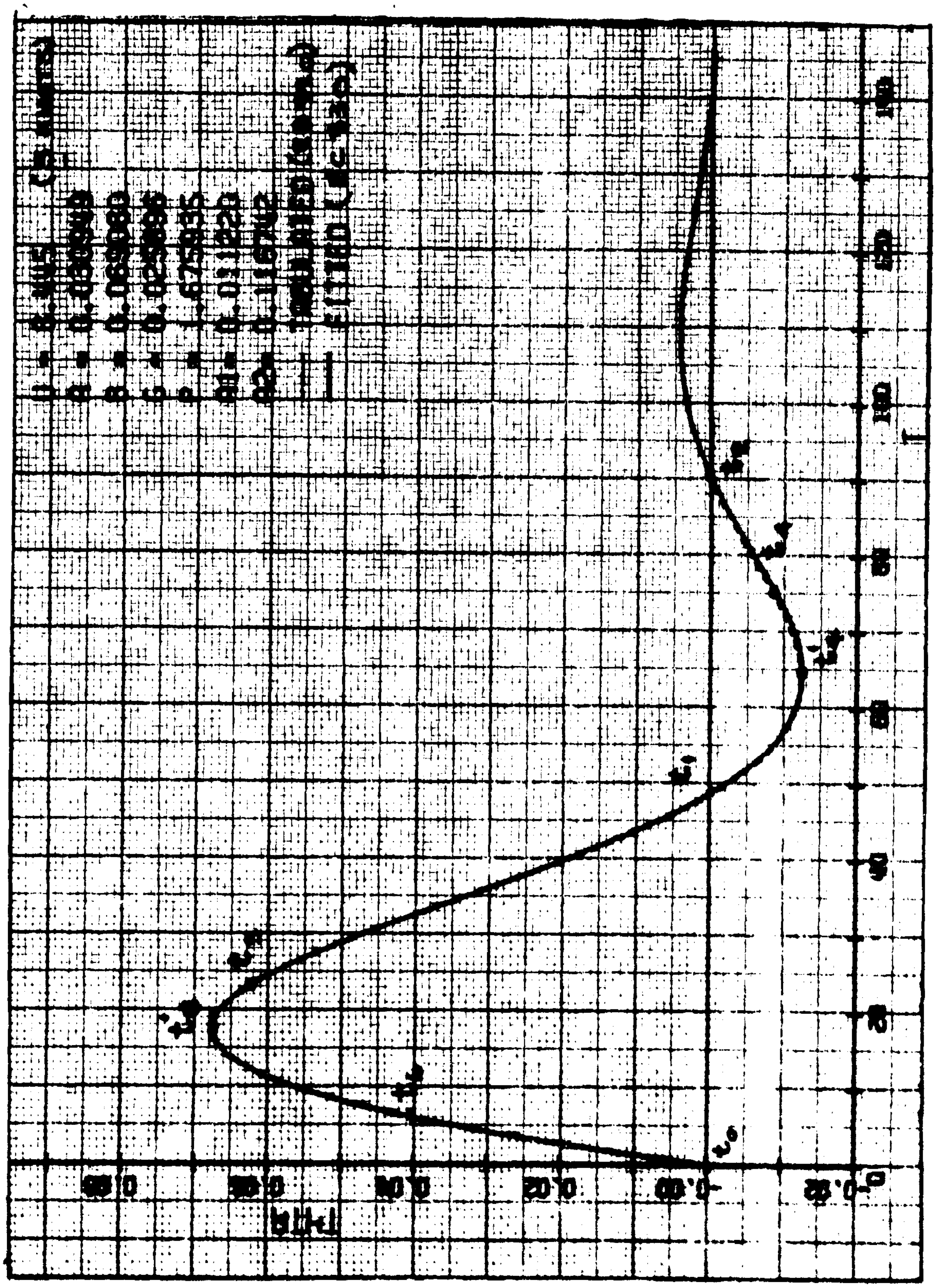


Figure 20. Submarine Response Curve, Longitudinal Case

ψ - at t_1, assuming a_1 term is zero:

$$\cos(\beta t_1 - \psi) = 0$$

$$\psi = \beta t_1 - \pi/2$$

α - assuming a_1 term is zero. Assume

$$t_3 = t_1 - P/2 \; ; \quad t_4 = t_1 + P/2$$

The computer will interpolate from the input data to find the ordinate values of points t_3 and t_4. Then

$$\frac{y_3}{y_4} = - \frac{a_2 e^{-\alpha t_3}}{a_2 e^{-\alpha t_4}}$$

$$e^{\alpha(t_4 - t_3)} = -y_3/y_4$$

$$\alpha = \frac{\ln(-y_3/y_4)}{t_4 - t_3}$$

a_2 - assume a_1 term is zero

$$a_2 = y_3 / e^{-\alpha t_3}$$

a_1

$$a_1 = -a_2 \cos(\psi)$$

γ - from equation (33)

$$y(6) = a_1 e^{-\gamma t(6)} + \underbrace{a_2 e^{-\alpha t(6)} \cos(\beta t(6) - \psi)}_{A1}$$

$$\gamma = -\frac{1}{t(6)} \ln\left(\frac{y(6) - A1}{A1}\right)$$

These calculated values are printed out as the initial estimates. Since the first estimate assumed the term a_1 was zero and we now have a value for this term, new estimates can be made.

$$APLN = \frac{y_3 - a_1 e^{-\gamma t_3}}{-y_4 + a_1 e^{-\gamma t_4}}$$

$$\alpha_{new} = \ln(APLN)/P$$

$$a_{2\,new} = \frac{y_3 - a_1 e^{-\gamma t_3}}{e^{-\alpha t_3}}$$

$$a_{1\,new} = -a_2 \cos\psi$$

$$\gamma_{new} = -\frac{1}{t(6)} \ln\left(\frac{y(6) - A1}{a_{1\,new}}\right)$$

These values are printed out in the second (improved) estimates along with the estimates of β and ψ made above.

The second estimates may not be adequate due to unsatisfactory mathematical assumptions. The values of a_1 and a_2 can be estimated a third way and used if they subsequently prove more accurate than 2nd Estimate values.

a_2

$$a_2 = y_3 / (-\cos\psi\ e^{-\gamma t_3} + e^{-\alpha t_3} \cos(\beta t_3 - \psi))$$

a_1

$$a_1 = -a_2 \cos\psi$$

These new, alternate, values of a_1 and a_2 are printed out as the third estimate along with the other printed values of the second estimate.

The values computed above (first choice should be the second estimate

output values) may be used in program EC310, Curve Fitting Program along with the data deck cards (T and THETA) from the impulse run on EB920 and originally used as inputs to the subject program, EC330. This program, EC310, will calculate and plot the values of y versus t from equation (33) for the estimated coefficients. It will also plot the original impulse data from EB920 as recorded on the punch card data. The first graphical recording of these two curves will show whether a satisfactory agreement of the equation (33) values (with newly estimated variables) and the equations of motion data are adequate. Figure 20 shows excellent convergence of these values.

If these curves do not overlay satisfactorily program EC310 may be rerun with the convergence option. The variables of equation (33) are optimized to convergence by Brown's routine. The function will sometimes converge and a new, satisfactory set of values (α, β, δ, etc.) will be calculated. If convergence is not achieved, additional calculations or intuitive estimates of closer values can be used. These can be resubmitted to EC310 for proof till convergence is reached.

2. INPUT DATA DECK

The input data consists principally of

a. Punched cards, cards of time and Theta from impulse run of EB920, Submarine Simulation Program, punched at two-second intervals.

b. Additional control data

The data deck format is given in table 19.

TABLE 19. INPUT DATA DECK, PROGRAM EC330

Card	Column(s)	Format	Description
1	1-5	I5	N, number of data points, Y(I) from EB920 run (one per punched card)
2	1-10	F10.5	U, submarine speed, knots
2	11-18	2A4	NAME, the name of the dependent variable, THETA
3-n	1-10	E15.7	THETA(I) values on punched cards from EB920, radians
3-n	11-20	E15.7	T, time values on punched cards from EB920, seconds
n+1			Blank card for normal end of job

3. OUTPUT DATA

The output data includes sequentially-numbered pages, each identified as EC330. The following terms are printed:

U - submarine speed, knots

T(I) and THETA(I) - all the values of the impulse run of EB920 as read by the card inputs to the subject program

A, B, G, P, A1, A2 - First estimates as described in equations 2 through 7 above.

A, B, G, P, A1, A2 - Improved estimates as noted in equations 8 through 11 above.

A, B, G, P, A1, A2 - Alternate estimates for a_1 and a_2 as described in equations 12 and 13 above.

4. LISTINGS AND FLOW CHART

The listings and flow chart for program EC330 are given in appendix A.

L. EC310, BROWN'S CONVERGENCE AND COMPARATIVE PLOT PROGRAM

1. DESCRIPTION

This program has the option of calculating and plotting the curve of impulse response in accordance with equation (24) or (33). It can, in addition, calculate the difference at each time, t_i, between a tabulated EB920 run and the calculated values of an EC310 program run; set up a series of non-linear equations that are the summations of partial derivatives of either equation (24) and (33) times the difference between the two ordinate values at t_i; and finally solve these non-linear equations by Brown's [2] routine. The values of α, β, γ, δ, ψ, a_1, a_2 and a_3 are optimized or converged by repeating the above solutions to limits within desired difference values (usually .0001) of successive estimates within the program. These converged values are then printed out along with graphs of the final solution of equation (24) or equation (33) and the original EB920 plot as in figure 21.

This program may have difficulties in convergence if eight variables are used. In this case the values of δ and a_3 are held fixed at initial input values.

The equations utilized in Brown's routine are the summations of the partial derivatives times the difference in Y (ordinate) values between the calculated and tabulated conditions. These are included below for clarity.

Let y_{T_i} be the value of the tabulated points (EB920 values on the punched data cards).

Assume the following function is to be fitted to the data points:

$$y_i = a_1 e^{-\gamma t_i} + a_2 e^{-\alpha t_i} \cos(\beta t_i + \psi) + a_3 e^{-\delta t_i} \qquad (34)$$

where α, β, γ, δ, ψ, a_1, a_2 and a_3 are to be determined by the method of least squares. The mean square error becomes

$$E = \frac{1}{N} \sum_{i=1}^{M} (y_i - y_{T_i})^2$$

Taking partial derivatives with respect to each parameter and equating to zero gives the following eight equations to be solved by Brown's routine. The routine will recalculate this till the n^{th} and $(n-1)^{th}$ term agree to a preset number of significant digits (called NUMSIG in this program).

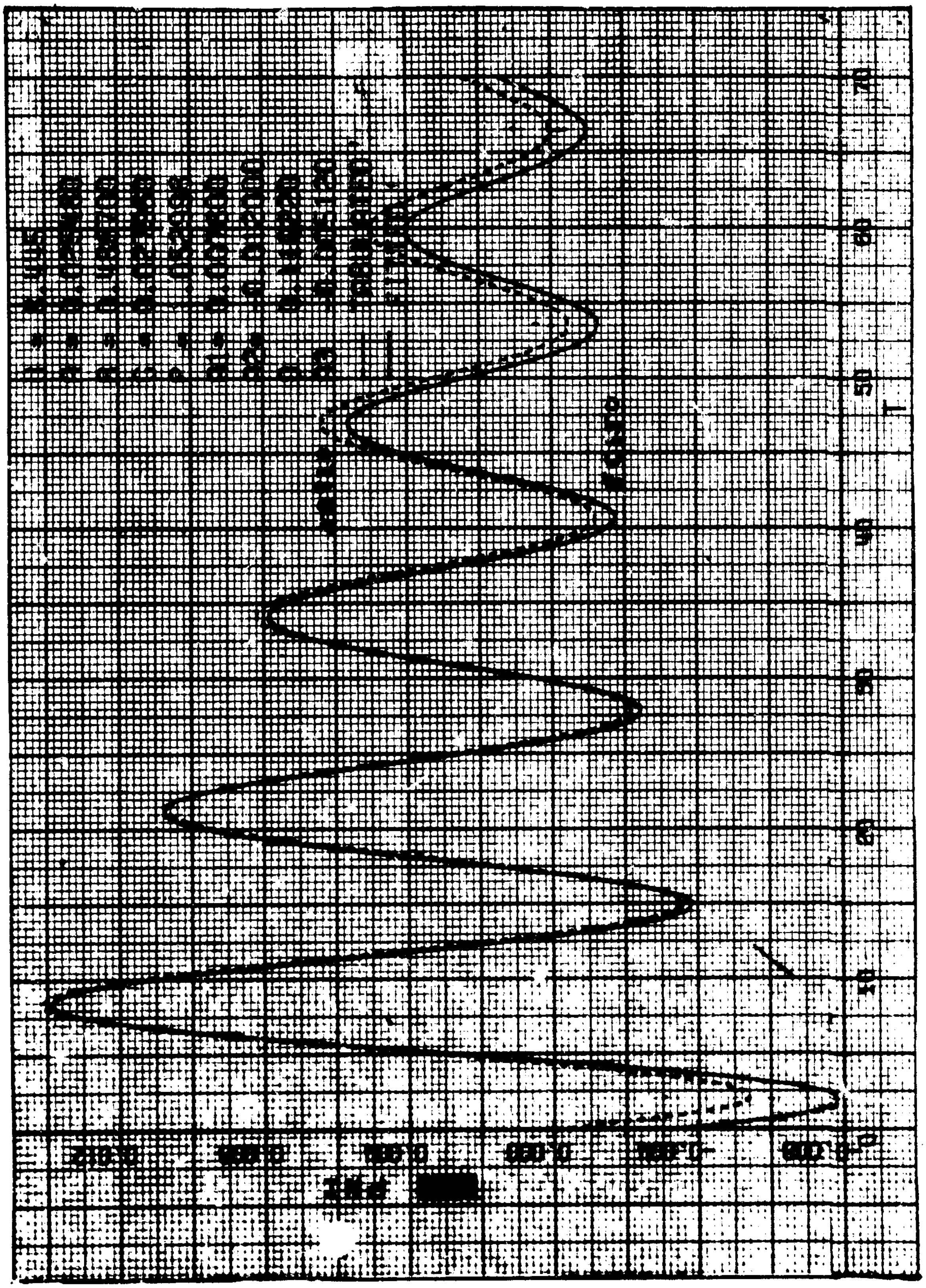


Figure 21. Typical Converged Response Plot

$$\alpha: \sum (y_i - y_{Ti})(-t_i a_3 e^{-\alpha t_i})$$
$$\beta: \sum (y_i - y_{Ti})(-t_i a_2 e^{-\alpha t_i} \sin(\beta t_i + \psi))$$
$$\gamma: \sum (y_i - y_{Ti})(-t_i a_1 e^{-\gamma t_i})$$
$$\psi: \sum (y_i - y_{Ti})(a_2 e^{-\alpha t_i} \sin(\beta t_i + \psi))$$
$$a_1: \sum (y_i - y_{Ti})(e^{-\gamma t_i})$$
$$a_2: \sum (y_i - y_{Ti})(e^{-\alpha t_i} \cos(\beta t_i + \psi))$$

The following two values can be added to those for requested optimizing, but usually will cause divergence rather than convergence:

$$\delta: \sum (y_i - y_{Ti})(-t_i a_3 e^{-\delta t_i})$$

$$a_3 \sum (y_i - y_{Ti})(e^{-\delta t_i})$$

Brown's routine then solves the above system of simultaneous, nonlinear equations. The algorithm used is quadratically convergent and requires only $(n^2/2 + 3n/2)$ function evaluations per iterative step as compared with $(n^2 + n)$ evaluations for Newton's Method. This results in a savings of computational effort for sufficiently complicated functions. A detailed description of the general method and proof of convergence are included in reference 2. Basically the technique consists in expanding the first equation in a Taylor series about the starting guess, retaining only linear terms, equating to zero and solving for one variable, say x_k, as a linear combination of the remaining n-1 variables. In the second equation, x_k is eliminated by replacing it with its linear representation found above, and again the process of expanding through linear terms, equating to zero and solving for one variable in terms of the now remaining n-2 variables is performed. One continues in this fashion, eliminating one variable per equation, until for the nth equation, we are left with one equation in one unknown. A single Newton step is now performed, followed by back-substitution in the triangularized linear system generated for the x_i's. A pivoting effect is achieved by choosing for elimination at any step that variable having a partial derivative of largest absolute value. The pivoting is done without physical interchange of rows or columns.

The vector of initial guesses, X, the number of significant digits desired, the maximum number of iterations to be used, and the number of equations to be solved are input data. After execution of the procedure, the vector x is the solution of the system (or best approximation thereto). A printout that a Jacobian-related matrix was singular is indicative of the process "blowing-up". If this occurs, try another initial estimate of values:

$$\alpha, \beta, \gamma, \psi, a_1, a_2, \delta \text{ and/or } a_3$$

This program can be used with either a rudder or elevator impulse. To find frequencies of oscillation in response to a rudder (lateral) impulse:

a. Run EB920 for the DR(lateral or rudder) impulse of desired magnitude, producing punched cards of output phi (Ø) and time (t) as well as printout data of dynamic response.

b. Exercise EC320, Root Cracker Program, 3D, Lateral to determine the roots of Equation (35), namely:

$$\alpha, \beta, \gamma \text{ and } \delta$$

c. Using the cards from EB920 and α, β, γ, and δ from EC320 as inputs, exercise program EC150, Time Response Coefficient Estimator, Lateral. This program provides estimated values of several terms of the lateral impulse equation:

$$y = a_1 e^{-\gamma t} + a_2 e^{-\alpha t} \cos(\beta t + \psi) + a_3 e^{-\delta t} \qquad (35)$$

where y represents bank angle, phi (Ø)

The values solved are:

a_1, a_2, a_3, and ψ

d. Using the coefficients and roots of b. and c. above, and the punch cards of EB920 of a. above, exercise program EC310. This program plots the tabulated data of the punch cards (dotted curve similar to figure 21) and the calculated response curve per equation 35 (solid curve of figure 21). A printout of the calculated and tabulated data is provided.

To find frequencies of oscillation in response to an elevator (longitudinal) impulse:

a. Run EB920, Submarine Simulation Program for the DS (longitudinal

or elevator) impulse of desired magnitude. The program output will supply a set of punched cards each carrying a value of theta (θ) and time (t) as well as the printout data of these same values. This data is the true response of the submarine to an elevator impulse.

b. Using these cards as input data, exercise program EC330, Time Response Coefficient Estimator, Longitudinal. This program provides estimated values for the constants of the longitudinal impulse equation:

$$y = a_1 e^{-\gamma t} + a_2 e^{-\alpha t} \cos(\beta t + \psi)$$

where y represents pitch angle, theta (θ)

The values solved are:

$$\alpha,\ \beta,\ \gamma,\ \psi,\ a_1 \text{ and } a_2$$

You may use the values α, β, and γ provided through program EC140, Root Cracker Program, 3D, Longitudinal.

c. Using these estimated roots and coefficients along with the punch card data from EB920 noted in a. above as inputs, exercise program EC310. This program provides a computer plot of the actual response curve provided through the punch cards of EB920 (dotted curve of figure 21), the calculated response curve per equation (33)(solid curve of figure 21). A printout of the calculated and tabulated data is also provided.

2. SUBROUTINE DESCRIPTIONS

PLOTS - plotting routine purchased from California Computer for use with their plotter. They must be secured from this company.

PLOT - plotting routine from Cal Comp, to start pen

FCNPLT - plot routine which labels axes, draws lines and symbols and calls for other subroutines

SCALE - plot routine from Cal Comp for automatic scaling of axes.

AXIS - plot routine from Cal Comp to produce axis

SYMBOL - plot routine from Cal Comp to produce letters and numbers on graph

NUMBER - plot routine from Cal Comp to write a floating point number from the program

LINE - plot routine from Cal Comp that connects a series of points

AUXFCN - this routine calculates the function of equations 1 or 2 and prepares the matrix of equations (Eq's 5 through 12) for Brown's routine (also known as subroutine SYSTEM in this program). This subroutine also calculates the magnitude of each of the equations 5 through 12.

SYSTEM - Brown's routine which solves six or eight (usually restrict the value to six in order to secure convergence) simultaneous, non-linear equations.

3. INPUT DATA DECK

The input data to this program include the punch card data from the particular impulse run on program EB920, control data for Brown's routine, plot control, and initial guesses or starting values of

$\alpha, \beta, \gamma, \psi, a_1, a_2$ - longitudinal

$\alpha, \beta, \gamma, \psi, a_1, a_2, \delta, a_3$ - lateral

These are identified in detail with their locations in table 20.

TABLE 20. INPUT DATA DECK, PROGRAM EC310

Card	Column(s)	Format	Description
1	1-5	I5	N, number of equations to be used N = 6 for longitudinal N = 6 or 8 for lateral (8 will seldom converge, so 6 is recommended)
1	6-10	I5	NUMSIG, number of significant digits of agreement between successive iterates which will cause convergence by the program. This value has been set at 4 for work on this program.
1	11-15	I5	MAXIT, maximum number of iterations to be performed by Brown's routine.
1	16-20	I5	IPRINT, will print out all iterations of values, up to MAXIT above if set to 1. IPRINT = 0 will print only first and

TABLE 20. INPUT DATA DECK, PROGRAM EC310 (cont.)

Card	Column(s)	Format	Description
			last values of values calculated ($\alpha, \beta, \gamma, \psi, a_1, a_2, \delta, a_3$)
1	21-25	I5	NPTS, number of data prints. This is the number of data cards from EB920.
1	26-30	I5	ISW1 = 1 - will calculate and print the equations (1 or 2) at the initial guess. ISWI = 0 - will attempt to converge for better values of $\alpha, \beta, \gamma, a_1, a_2, \psi$ with Brown's routine and calculate equations 1 or 2 for the converged values.
1	31-35	I5	IPLOT = 1 - will plot the tabulated and calculated data IPLOT = 0 - no plot
2	1-10	F10.5	U, forward speed, knots
2	11-18	2A4	IY, word "PHI" for lateral case word "THETA" for longitudinal case
3-n	1-15	E15.7	Y(I), values of dependent variable (phi or theta) from cards of EB920 run, I = NPTS
3-n	16-30	E15.7	T(I), values of independent variable (t) from cards of EB920.
u+1	1-80	8F10.5	A, B, G, P, A1, A2, D, A3 ($\alpha, \beta, \gamma, \psi, a_1, a_2, \delta, a_3$), initial estimates of unknowns of equations 1 or 2.
m+2	-	-	Blank card for normal end of job.

4. OUTPUT DATA

Printout Data:

a. Output values - The printout across the page will include columns of:

N^{th} time through Brown's routine - 0 = initial estimate

$\alpha, \beta, \gamma, \psi, a_1, a_2$

b. Value of partial derivatives of equation 1 at convergence in a row of the following order:

$$\frac{\partial y}{\partial \alpha}, \frac{\partial y}{\partial \beta}, \frac{\partial y}{\partial \gamma}, \frac{\partial y}{\partial \psi}, \frac{\partial y}{\partial a_1}, \frac{\partial y}{\partial a_2}$$

c. Converged, or final values of calculated terms in a row in following order:

$\alpha, \beta, \gamma, \psi, a_1, a_2, \delta, a_3$

d. Columns of time (T), and the calculated and tabulated values of Y(I) in the following order:

T YE YT

Graphical Data:

If the CAL COMP subroutines are available a plot as shown in figure 21 can be produced.

5. LISTINGS AND FLOW CHART

The listings and flow chart for program EC310 are given in appendix A.

M. EC790, CALCULATION OF COMPACT COEFFICIENTS

1. DESCRIPTION

Program EB920, Submarine Simulation Program, was written to provide a flexible, analytical tool for examining submarine performance with a number of optional devices of submarine control, program control and graphical as well as printed output. To provide the computer control necessary for a submarine training device, the many optional controls and outputs of EB920 are not required; a number of the coefficients and constants can be combined to require fewer multiplications and other operations; and a much smaller computer can be used. This program uses the normal input coefficients, submarine physical constants, and control values of program EB920 and the exact input data deck from an EB920 run may be used. The program then combines a number of coefficients, multiplies and divides them by the appropriate constants. This program thus, takes data identical to that for a scientific run with program EB920; combines coefficients and multiplies by constants, prints the original and combined coefficients separately; and finally punches data cards for input to program EC780 compact submarine simulation program when it has been compiled on a small computer. Use of this smaller computer demonstrates that a much smaller machine (8K core) can provide adequate storage for real time control of an actual training device.

Equation group number (36) shows the mathematical operations performed on the original coefficients and constants used for the inputs to program EB920. The primed values on the right side of the equation are identical to those used in EB920 and are defined in the glossary. The unprimed values on the left side of the equation are repeated as FORTRAN variables used by program EC790 and EC780.

2. INPUT DATA DECK

The program requires data input exactly as required by program EB920. All units coefficients, and constants, etc., as used in an EB920 run are identical. This input data deck assembly and use has been described in that program.

Since EC780, which will use the output deck from this program, cannot perform many of the operation of EB920, some of the data normally required in EB920 runs can be left blank in the proper punch locations on the data deck if desired. However, all cards required by EB920, even if completely blank, must be present as input to this program.

3. OUTPUT DATA

Two data sheets are printed by this program. The first one is identical to figure of program EB920 and shows the original coefficients present on the input data deck. The second one is printed out in the same format but the values of those coefficients to be used by program EC780 are printed instead. The other coefficients that were not modified in this program are printed on this same page, but are not used further.

Coefficient Description (EC780) (36)

$$XDRDR' = X'_{\delta r \delta r} = \frac{\rho}{2} \ell^2 X_{\delta r \delta r}$$

$$XDSDS' = X'_{\delta s \delta s} = \frac{\rho}{2} \ell^2 X_{\delta s \delta s}$$

$$XDBDB' = X'_{\delta b \delta b} = \frac{\rho}{2} \ell^2 X_{\delta b \delta b}$$

$$A11' = a'_{i1} = \frac{\rho}{2} \ell^2 a_{i1}$$

$$A12' = b'_{i1} = \frac{\rho}{2} \ell^2 b_{i1}$$

$$A13' = c'_{i1} = \frac{\rho}{2} \ell^2 c_{i1}$$

$$A21' = a'_{i2} = \frac{\rho}{2} \ell^2 a_{i2}$$

$$A22' = b'_{i2} = \frac{\rho}{2} \ell^2 b_{i2}$$

$$A23' = c'_{i2} = \frac{\rho}{2} \ell^2 c_{i2}$$

A31'	=	a'_{i3}	=	$\frac{\rho}{2} \ell^2 a_{i3}$
A32'	=	b'_{i3}	=	$\frac{\rho}{2} \ell^2 b_{i3}$
A33'	=	c'_{i3}	=	$\frac{\rho}{2} \ell^2 c_{i3}$
XUD'	=	$X'_{\dot{u}}$	=	$\frac{\rho}{2} \ell^3 X_{\dot{u}}$
YR'	=	Y'_r	=	$(\frac{\rho}{2} \ell^3 Y_r - m)$
YRD'	=	$Y'_{\dot{r}}$	=	$\frac{\rho}{2} \ell^4 Y_{\dot{r}}$
YPD'	=	$Y'_{\dot{p}}$	=	$\frac{\rho}{2} \ell^4 Y_{\dot{p}}$
YP'	=	Y'_p	=	$\frac{\rho}{2} \ell^3 Y_p$
YV'	=	Y'_v	=	$\frac{\rho}{2} \ell^2 Y_v$
YVAV'	=	$Y'_{v\lvert v\rvert}$	=	$\frac{\rho}{2} \ell^2 Y_{v\lvert v\rvert}$
YDR'	=	$Y'_{\delta r}$	=	$\frac{\rho}{2} \ell^2 Y_{\delta r}$
YVD	=	$Y'_{\dot{v}}$	=	$\frac{\rho}{2} \ell^3 Y_{\dot{v}}$
ZQ'	=	Z'_q	=	$\frac{\rho}{2} \ell^3 Z_q$
ZQD'	=	$Z'_{\dot{q}}$	=	$\frac{\rho}{2} \ell^4 Z_{\dot{q}}$
ZRR'	=	Z'_{rr}	=	$\frac{\rho}{2} \ell^4 Z_{rr}$

$$ZVR' = Z'_{vr} = \frac{\rho}{2}\ell^3 Z_{vr}$$

$$ZSTR' = Z'_{*} = \frac{\rho}{2}\ell^2 Z_{*}$$

$$ZW' = Z'_{w} = \frac{\rho}{2}\ell^2 Z_{w}$$

$$ZWAW' = Z'_{w|w|} = \frac{\rho}{2}\ell^2 Z_{w|w|}$$

$$ZVV' = Z'_{vv} = \frac{\rho}{2}\ell^2 Z_{vv}$$

$$ZDS' = Z'_{\delta s} = \frac{\rho}{2}\ell^2 Z_{\delta s}$$

$$ZDB' = Z'_{\delta b} = \frac{\rho}{2}\ell^2 Z_{\delta b}$$

$$ZWD' = Z'_{\dot{w}} = \frac{\rho}{2}\ell^3 Z_{\dot{w}}$$

$$T_4 = 1/(I_x - \frac{\rho}{2}\ell^5 K_{\dot{p}})$$

$$AKRD' = K'_{\dot{r}} = \frac{\rho}{2}\ell^5 K_{\dot{r}} \cdot T_4$$

$$AKP' = K'_{p} = \frac{\rho}{2}\ell^4 K_{p} \cdot T_4$$

$$AKVD' = K'_{\dot{v}} = \frac{\rho}{2}\ell^4 K_{\dot{v}} \cdot T_4$$

$$AKV' = K'_{v} = \frac{\rho}{2}\ell^3 K_{v} \cdot T_4$$

$$AKVAV' = K'_{v|v|} = \frac{\rho}{2}\ell^3 K_{v|v|} \cdot T_4$$

$$AKPD' = K'_{\dot{p}} = T_4$$

$$T_s = 1/(I_y - \frac{\rho}{2} \ell^5 M_{\dot{q}})$$

$$AMRP' = M'_{rp} = (I_z - I_x + \frac{\rho}{2} \ell^5 M_{rp}) \cdot T_s$$

$$AMRR' = M'_{rr} = \frac{\rho}{2} \ell^5 M_{rr} \cdot T_s$$

$$AMWD' = M'_{\dot{w}} = \frac{\rho}{2} \ell^4 M_{\dot{w}} \cdot T_s$$

$$AMVR' = M'_{vr} = \frac{\rho}{2} \ell^4 M_{vr} \cdot T_s$$

$$AMQ' = M'_q = \frac{\rho}{2} \ell^4 M_q \cdot T_s$$

$$AMAWQ' = M'_{|w|q} = \frac{\rho}{2} \ell^4 M_{|w|q} \cdot T_s$$

$$AMSTR' = M'_* = \frac{\rho}{2} \ell^3 M_* \cdot T_s$$

$$AMW' = M'_w = \frac{\rho}{2} \ell^3 M_w \cdot T_s$$

$$AMWAW' = M'_{w|w|} = \frac{\rho}{2} \ell^3 M_{w|w|} \cdot T_s$$

$$AMVV' = M'_{vv} = \frac{\rho}{2} \ell^3 M_{vv} \cdot T_s$$

$$AMOS' = M'_{\delta s} = \frac{\rho}{2} \ell^3 M_{\delta s} \cdot T_s$$

$$AMDB' = M'_{\delta b} = \frac{\rho}{2} \ell^3 M_{\delta b} \cdot T_s$$

$$AMQD' = M'_{\dot{q}} = T_s$$

$T_6 \quad = \quad 1/(I_z - \frac{\rho}{2} \ell^5 N_{\dot{r}})$

$ANPQ' \quad = \quad N'_{pq} \quad = \quad (I_x - I_y + \frac{\rho}{2} \ell^5 N_{pq}) \cdot T_6$

$ANPD' \quad = \quad N'_{\dot{p}} \quad = \quad \frac{\rho}{2} \ell^5 N_{\dot{p}} \cdot T_6$

$ANVD' \quad = \quad N'_{\dot{v}} \quad = \quad \frac{\rho}{2} \ell^4 N_{\dot{v}} \cdot T_6$

$ANP' \quad = \quad N'_p \quad = \quad \frac{\rho}{2} \ell^4 N_p \cdot T_6$

$ANR' \quad = \quad N'_r \quad = \quad \frac{\rho}{2} \ell^4 N_r \cdot T_6$

$ANV' \quad = \quad N'_v \quad = \quad \frac{\rho}{2} \ell^3 N_v \cdot T_6$

$ANVAV' \quad = \quad N'_{v|v|} \quad = \quad \frac{\rho}{2} \ell^3 N_{v|v|} \cdot T_6$

$ANDR' \quad = \quad N'_{\delta r} \quad = \quad \frac{\rho}{2} \ell^3 N_{\delta r} \cdot T_6$

$ANRD' \quad = \quad N'_{\dot{r}} \quad = \quad T_6$

$ZB' \quad = \quad z'_B \quad = \quad CB \cdot z_B$

This program also punches the newly calculated, modified coefficients on data cards for direct use in EC780, Compact Submarine Simulation Program. This data is punched in 6E13.6 format on the data cards. The terms are identified in table 21.

TABLE 21. OUTPUT DECK FORMAT, EC790

Card	Variables
1	XDRDR, XDSDS, XDBDB, A11
2	A12, A13, A21, A22, A23, A31
3	A32, A33, XUD, YR
4	YRD, YPD, YP, YV, YVAB, YDR
5	YVD, ZQ, ZQD, ZRR
6	ZVR, ZSTR, ZW, ZWAW, ZVV, ZDS
7	ZDB, ZWD, AKRD, AKP, AKV
8	AKV, AKVAV, AKPD, AMRP, AMRR
9	AMWD, AMVR, AMQ, AMAWQ, AMSTR, AMW
10	AMWAW, AMVV, AMDS, AMDB, AMQD
11	ANPQ, ANPD, ANVD, ANP, ANR, ANV
12	ANVAV, ANDR, ANRD, DRMAX, ETAHI
13	ETALO, CW, CB, XG, ZG, AL
14	AM, DR, DS, DB, ZB, UC
15	TIME, R1, DELTMA, SWMAX, R2, DELTMI
16	DSF, DRF, Y(1), Y(2), Y(3), Y(4), Y(5), Y(6), Y(7), Y(8), Y(9), Y(10), Y(11), Y(12)

4. LISTINGS AND FLOW CHART

The listings and flow chart for program EC790 are given in appendix A.

N. EC780, COMPACT SUBMARINE SIMULATION PROGRAM

1. DESCRIPTION

This program is a limited, modified version of EB920, Submarine Simulation Program. It will provide all output data in accordance with the equations of motion. A number of types of preplanned submarine maneuvers have been included for testing the program. This program has been modified from EB920 by utilizing a number "synthetic coefficients" which were calculated in program EC790, Calculation of Compact Coefficients. These synthetic coefficients are the normal coefficients from the equations of motion (as used directly in program EB920) that have been combined and multiplied by appropriate constants in this separate program rather than the simulation program. These new coefficients, along with the removal of a number of options from the scientific research program, EB920, allow this program, EC780, to solve the equations of motion with fewer operations and much less core.

This program operates with the modified coefficients from program EC790. The specific mathematical model is therefore somewhat different from that described in Reference 1 and used in program EB920, and are given in equations (37) through (42).

Coefficient terms used above are defined in the associated input data program, EC790.

2. SUBROUTINE DESCRIPTIONS

CONTR - This subroutine allows control of submarine motion by programed movement of elevators and/or rudder. This allows selection of the various maneuvers:

a. Steady dive, turn, or combination
b. Meander or overshoot
c. Flat turn with autopilot
d. Climbing turn, combination of a programed turn and meander or overshoot
e. DS or elevator impulse (no punched cards are prepared by the computer as in this maneuver with program EB920) with autopilot
f. DR or rudder impulse (no punched cards)
g. Acceleration/deceleration
h. Maximum acceleration/deceleration

These controls function and are input exactly as in program EB920, so reference is made to this program for detailed information.

UPDATE - This subroutine sets the propeller thrust constants, ai, bi, ci in accordance with the current value of η ; (u_c / u). It solves the equations of motion at each updated time increment, H. These values, $\dot{u}$, $\dot{v}$,

Compact Mathematical Model
Submarine Equations of Motion

Axial (37)

$$
\begin{aligned}
\dot{u} = \Big[& m\big(vr - wq \\
& + X_G(q^2 + r^2) - Z_G(pr + \dot{q})\big) \\
& + u^2\left(X_{\delta r \delta r}\delta_r^2 + X_{\delta s \delta s}\delta_s^2 + X_{\delta b \delta b}\delta_b^2 + a_i\right) \\
& + u_c\left(b_i u + c_i u_c\right) \\
& - (W - B)\sin\theta \Big] \Big/ \left[m - X_{\dot{u}}\right]
\end{aligned}
$$

Lateral (38)

$$
\begin{aligned}
\dot{v} = \Big[& u\left(Y_{ur} r + Y_p p + Y_v v\right) \\
& - m\big(Z_G(qr - \dot{p}) + X_G(qp + \dot{r}) - wp + ur\big) \\
& + Y_{\dot{r}}\dot{r} + Y_{\dot{p}}\dot{p} + Y_{v|v|} v (v^2 + w^2)^{\frac{1}{2}} \\
& + Y_{\delta r} u^2 \delta_r + (W - B)\cos\theta \sin\phi \Big] \Big/ \left[m - Y_{\dot{v}}\right]
\end{aligned}
$$

Normal (39)

$$\dot{w} = \Big[Z_{vp} v p + u(Z_q q + Z_w w) - m\big(Z_G(p^2 + q^2) - X_G(rp - \dot{q}) + qu - vp\big) + Z_{\dot{q}}\dot{q} + Z_{rr} r^2 + Z_{vr} v r + Z_{vv} v^2 + Z_{w|w|} w (v^2 + w^2)^{\frac{1}{2}} + u^2(Z_{\delta s}\delta_s + Z_{\delta b}\delta_b + Z_*) + (W - B)\cos\theta\cos\phi\Big] \Big/ \Big[m - Z_{\dot{w}}\Big]$$

Roll (40)

$$\dot{p} = m\; Z_G(\dot{v} - wp + ur)K_{\dot{p}} + K_{\dot{r}}\dot{r} + K_{\dot{v}}\dot{v} + u(K_p p + K_v v) + K_{v|v|} v (v^2 + w^2)^{\frac{1}{2}} - K_{\dot{p}}(Z_G W - Z_B B)\cos\theta\cos\phi$$

Pitch (41)

$$\dot{q} = M_{rp} rp + M_{\dot{w}}\dot{w} + u(M_q q + M_w w) - m\big(Z_G(\dot{u} - vr + wq) - X_G(\dot{w} - uq + vp)\big)M_{\dot{q}} + M_{rr} r^2 + M_{vr} v r + M_{vv} v^2 + M_{|w|q}\, q (v^2 + w^2)^{\frac{1}{2}} + M_{w|w|} w (v^2 + w^2)^{\frac{1}{2}} + u^2(M_* + M_{\delta b}\delta_b + M_{\delta s}\delta_s) - M_{\dot{q}}\big(X_G W\cos\theta\cos\phi + (Z_G W - Z_B B)\sin\theta\big)$$

Yaw (42)

$$\dot{r} = N_{pq} pq + N_{\dot{p}}\dot{p} + N_{\dot{v}}\dot{v} + u^2 N_{\delta r}\delta_r - m\big(X_G(\dot{v} - wp + ur)\big)N_{\dot{r}} + N_{v|w|} v (v^2 + w^2)^{\frac{1}{2}} + u(N_p p + N_r r + N_v v) + N_{\dot{r}}(X_G W)\cos\theta\sin\phi$$

$\dot{w}$, $\dot{p}$, $\dot{q}$, $\dot{r}$, with kinematics, $\dot{\Theta}$, $\dot{\Psi}$, $\dot{\phi}$, and $\dot{x}$, $\dot{y}$, and $\dot{z}$ are then integrated over the time interval, H. These new values returned to the main program for possible printout, and the old derivative values replaced by the new values each pass through the subroutine.

3. INPUT DATA DECK

The input data consists of two special control cards plus the punch card output data deck from EC790, Calculation of Compact Coefficients. The inputs to that program are exactly as for inputs to EB920 with the exception that values for terms not used may be left blank. The subject program is set up so that an input data deck that had been used for a particular run on EB920 can be used for input to EC790 and the output cards of EC790 used with the two special control cards as the input data deck for the subject program, EC780. The data deck format for this program is given in table 22.

TABLE 22. INPUT DATA DECK, PROGRAM EC780

Card	Column(s)	Format	Description
1	1-5	IS	IRUN, run number
1	6-10	IS	NPNT, calculated values will printout each NPNT integration cycle (If H = .25 seconds, and NPNT = 8, printout interval is NPNT x H = 2 seconds)
1	11-15	IS	NS. This variable selects the type of submarine control in CONTR subroutine: NS = 0, Fixed controls per initial conditions NS = 1, Overshoot, meander, etc. NS = 2, Special climbing term NS = 3, Flat turn (with autopilot) NS = 4, Elevator impulse NS = 5, Rudder impulse (with autopilot) NS = 6, Acceleration/deceleration (with autopilot) NS = 7, Maximum acceleration/deceleration (with autopilot) (Details of these controls are in EB920 write up)
2	11-20	F10.5	H, integration time increment, seconds. This step size is used throughout the entire run.

TABLE 22. INPUT DATA DECK, PROGRAM EC780 (cont.)

Card	Column(s)	Format	Description
2	21-30	F10.5	TLIM, time at the end of the run, seconds.
3-18	1-78	6E13.6	Punch card data from EC790, Calculation of compact coefficients. Table defines the variables punched on these cards.
19			Blank card for normal end of job.

4. OUTPUT DATA

A sample output data sheet from this program (using trial or synthetic input coefficients from EC790) is shown in figure 23. The run number and control routine value NS are provided in the first line of data. Then, at the desired intervals of time, the calculated values (and time) from the solutions to the equations of motion are printed (u, v, w, p, q, r, θ, Ψ, ϕ, x, y, z, and t).

5. COMPUTER RUN TIME

For the SDS Sigma 2 computer, cycle and add time is 2.25 microseconds. The subject programs required 0.2 seconds for complete integration period with no I/O.

6. LISTINGS AND FLOW CHART

The program is written in basic FORTRAN, primarily for use on an SDS, Sigma 2 or other small computers. It is a streamlined version of program EB920, and has operated on the IBM 360/40 computer. It is not intended for normal use on this larger computer as the input has been tailored for card-reader input and eighty-column typewriter output available on the Sigma 2. The listings and flow chart are given in appendix A.

SUBMARINE SIMULATION

RUN NO 2 NS 0

U P Y	V THETA Z	W PSI T	P PHI	Q X
0.344500E 01	0.0	0.435340E-01	0.0	0.0
0.0	0.515590E-02	0.0	0.0	0.0
0.0	0.800000E 03	0.0		
0.749529E 01	0.0	0.429612E-01	0.0	-0.311090E-04
0.0	0.501081E-02	0.0	0.0	0.806341E 02
0.0	0.800022E 03	0.100000E 02		
0.663187E 01	0.0	0.412510E-01	0.0	-0.656057E-04
0.0	0.452187E-02	0.0	0.0	0.151214E 03
0.0	0.800101E 03	0.200000E 02		
0.583151E 01	0.0	0.389526E-01	0.0	-0.875416E-04
0.0	0.374149E-02	0.0	0.0	0.213483E 03
0.0	0.800238E 03	0.300000E 02		
[illegible]	0.0	0.365288E-01	0.0	-0.905779E-04
0.0	0.28[illegible]483E-02	0.0	0.0	0.268024E 03
0.0	0.800431E 03	0.400000E 02		

Figure 22. Output Data Format, EC780

O. EC572 - WAVE GENERATOR

1. DESCRIPTION

Program EC572, Wage Generator Program, was written to provide a means of generating a random ocean wave surface at one point in time. It can be used with the Submarine Simulation Program to affect submarine motion close to the surface; but at present the outputs are only a choice of printed points, a graph plotted against time, or points and time punched on cards for PSD analysis. The program uses a mathematical model developed in "Mathematical Generation of a Realistic Sea", Hydro-autics, Inc. Technical Report 001-13 (DDC # AD 609906)[3] prepared for the Bureau of Ships in October 1963. However, the wave spectra used is not the standard Neumann used in the report, but the more up-to-date Pierson-Moskowitz spectra.[4] This spectra has the equations

$$A^2(\omega) = \frac{8.1\times10^{-3}\, g^2}{\omega^5}\, e^{-.74\left(\frac{g}{U\omega}\right)^4}$$

$A^2(\omega)$ = wave spectra

ω = frequency

g = gravity

U = wind speed

The program first determines the frequency points at which the area under $A^2(\omega)$ can be divided into equal sections. This is done in closed form by integrating $A^2(\omega)$ and finding the area between two limits.

$$Area = \int_A^B A^2(\omega)\, d\omega = \left. \frac{8.1\times10^{-3}\, U^4}{2.96\, g^2}\, e^{-.74\left(\frac{g}{U\omega}\right)^4} \right|_A^B$$

The total area is divided into the number of specified bands and the program solves for the upper limit of the expression to give this value. This value is then used as a lower limit for the next value until B is reached. The series of frequencies are stored. The program then calls a random number generator in the system library to generate enough random numbers ϵ_{ij} between zero and 2π for use in the expression below.

The surface amplitude is then calculated by means of the expression

$$\bar{\varphi}(t) = B \sum_{i=1}^{5} \sum_{j=1}^{n} (\omega_j)^{-5} e^{-.74\left(\frac{g}{U}\right)^4 (\omega_j)^4} \cos(\omega_j t + \epsilon_{ij}) (a_i \Delta\omega_j)^{1/2}$$

when

$$a_1 = a_5 = 0.9058$$
$$a_2 = a_4 = 0.43305$$
$$a_3 = 0.53254$$

$$B = \frac{2}{\pi} 8.1 \times 10^{-3} g^2$$

U is the speed in knots

Development of this equation is covered in the reference [3]. The rest of the program consists of control for inputs, outputs, error messages, and times the program is to be run.

2. SUBROUTINE DESCRIPTIONS

The CAL COM subroutines must be supplied by the operating system if plots are desired. They are

PLOTS	LINE	SCALE	IHEX
INIT	PLOT	AXIS	

Their function is described in program EC310. Other subroutines needed are

RANDM — Random number generator. Returns a floating point fraction between 0 and 1.

EXP
ALOG
SQRT — Standard FORTRAN calls
COS
EXIT

3. INPUT DATA DECK

The inout data deck format is given in table ?3.

TABLE 23. INPUT DATA DECK, PROGRAM EC572

Card	Column(s)	Format	Description
1	1-5	I5	NBAND - Number of energy bands in spectrum
	6-10	I5	NT - Number of times points are to be calculated (time)
	11-15	I5	NPLT - 0 = No Plot, 1 = Plot
	16-20	I5	NPCH - 0 = No Card Output, 1 = Card Output
	21-30	F10.5	A - Lower frequency limit (radians)
	31-40	F10.5	B - Upper frequency limit (radians)
	41-50	F10.5	DT - Seconds between each time of calculation
2	1-80	8F10.8	VI - The speed (U) at which the program is to be run. Any number of speeds up to eight can be used. Program will calculate only the number on this card.

4. OUTPUT DATA

a. Printed

a. Printed	NBAND	number of energy bands in spectrum
	NT	number of points
	NPLT	plot control
	NPCH	punch control
	A	lower frequency limit
	B	upper frequency limit
	DT	time interval
	V	wind velocity, knots
	U	wind velocity, ft/sec.
	GOU	g/U
	FACT1	$\frac{8.1 \times 10^{-3} U^4}{2.96 g^2}$
	FACT2	$-0.74(g/U)^4$
	YO	area below lower limit
	AY	area below upper limit
	AREA	area between limits
	AR	area in energy band
	X1	$g/U(-0.74/\ln(YO))^{\frac{1}{4}}$
	Y1	frequency limit of each band (X1·FACT1)

YO	area below each frequency band limit
TT	time
TSUM	

b. Plotted. The graph given in figure 24 is an example of the plotted output with

NBAND	=	10
NT	=	400
NPLT	=	1
NPCH	=	0
A	=	0.2
B	=	1.8
DT	=	1.0
U	=	23.7 kts. (40 ft/sec.)

If CALCOMP plotter software is not available, the plotting calls must either be removed from the source deck or a dummy subroutine deck must be used.

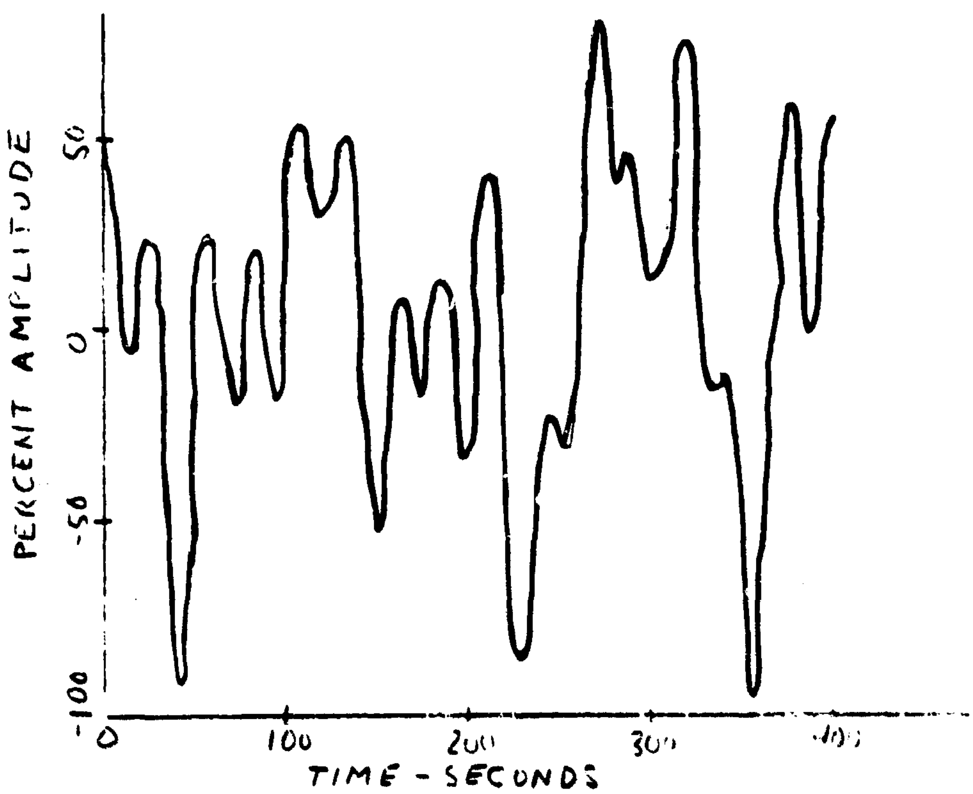


Figure 23. Typical Wave Generator Output

c. Cards. If NPCH is set equal to 1, a card deck is punched with identifying data on the first card followed by cards with the card number and value of $\xi(t)$ in a I5, 5x, F10.4 format. This output is for use in a PSD program.

d. A number of error messages and plot instructions are printed out on the computer operator's console to assist in running the program. If some other device number is used at another installation, the correct one will have to be used.

Device Number	Device
1	Card Reader
3	Line Printer
2	Card Punch
15	Operator's Typewriter

5. LISTINGS AND FLOW CHART

The listings and flow chart for program EC572 are given in appendix A.

REFERENCES

1. Gertler, M. and Hagen, G., "Standard Equations of Motion for Submarine Simulation", Research and Development Report 2510, Naval Ship Research and Development Center, Washington, D.C., June 1967.

2. Brown, K.M. and Conte, S.D., "The Solution of Simultaneous Non-Linear Equations", Proceedings of the 22nd. National Conference of the Association for Computing Machinery, 1967.

3. "Mathematical Generation of a Realistic Sea", Chen et al., Hydroautics, Inc., Technical Report 001-13, Bureau of Ships, October 1963, DDC # AD609906.

4. "Recent Developments in Seakeeping Research and Its Application to Design", Abkowitz et al., Proceedings of the Society of Naval Architects and Marine Engineers, New York, November 1966.

APPENDIX A
PROGRAMING LISTINGS AND FLOW CHARTS

This appendix contains the listings and flow charts for each of the computer programs described in this report. The page number for the start of each program is given in table 24.

TABLE 24. PROGRAM LISTINGS PAGE NUMBER

Program	Page	Program	Page	Program	Page	Program	Page
EB920	127	ZC300	235	EC320	277	EC790	315
ZC790	193	ZC690	240	EC150	283	EC780	328
EC470	226	ZC691	244	EC330	287	EC572	353
EC430	231	EC140	248	EC310	294		

The programs are written in FORTRAN IV and should run on any computer equiped with this complier. The higher level features of this language are not used and its programs are written to be as machine independent as possible. Each subroutine starts on a new page and the total program listing is followed by flow charts for that program.

The flow charts were generated from the FORTRAN listings. The program number, subroutine name, and page number of the flow chart are listed at the top of each page. The page number is used to connect the various charts together. Each input and output is labled with a decimal number. The number to the left of the decimal point is the page number (upper right-hand corner) of the connection and the right-hand part gives the box number on that page. Subroutine calls are given in the subroutine box in the same manner. All flow chart symbols are convential.

```
//        JOB   EB920
//        EXEC FFORTRAN
      DIMENSION Y(13). TL(12)
      DIMENSION SAVE(300,16), ILOC(16), BUFF(3000)
      REAL IX,IY,IZ,IXY,IXZ,IYZ
C
      COMMON H, HMAX, HMIN, DH, FCT, TL, NGS, N, IS1, NPNT
C
      COMMON TLIM, RHOL2, RHOL3, RHOL4, RHOL5, WMB, ETA, ETAM1, ISW2
C
      COMMON        XQQ, XRR, XRP, XUD, XVR ,XWQ, XUU, XVV,
     1              XWW, XDRDR, XDSDS, XDBDB, XVVE, XWWF, XDRDRE, XDSDSE
C
      COMMON        YRD, YPD, YPAP, YPQ, YQR, YVD, YVQ, YWP,
     1              YWR, YR, YP, YARDR, YVAR, YSTR, YV, YVAV,
     2              YVW, YDR, YRE, YVF, YVAVE, YDRE
C
      COMMON        ZQD, ZPP, ZRR, ZRP, ZWD, ZVR, ZVP, ZQ,
     1              ZAQDS, ZWAQ, ZSTR, ZW, ZWAW, ZAW, ZWW, ZVV,
     2              ZDS, ZDB, ZQE, ZWE, ZWAWF, ZDSE
C
      COMMON        AKPD, AKRD, AKQR, AKPQ, AKPAP, AKP, AKR, AKVD,
     1              AKVQ, AKWP, AKWR, AKSTR, AKV, AKVAV, AKVW, AKDR,
     2              AKSTRF
C
      COMMON        AMQD, AMPP, AMRR, AMRP, AMQAQ, AMWD, AMVR, AMVP,
     1              AMQ, AMAQDS, AMAWQ, AMSTR, AMW, AMWAW, AMAW, AMWW,
     2              AMVV, AMDS, AMDB, AMQE, AMWF, AMWAWE, AMDSE
C
      COMMON        ANRD, ANPD, ANPQ, ANQR, ANRAR, ANVD, ANWR, ANWP,
     1              ANVQ, ANP, ANR, ANARDR, ANAVR, ANSTR, ANV, ANVAV,
     2              ANVW, ANDR, ANRE, ANVE, ANVAVF, ANDRE
C
      COMMON         IX, IY, IZ, IXY, IXZ, IYZ
C
      COMMON         CW, CB, UC, XB, YB, ZB
C
      COMMON         DR, DS, DB, RHO, AL, AM
C
      COMMON        DRMAX, ETAHI, ETALO, A11, A12, A13
C
      COMMON        A21, A22, A23, A31, A32, A33
C
      COMMON        XG, YG, ZG
C
      COMMON        ILOC, IPLOT, IRUN, IOPEN, NPLT,IOPT
C
      COMMON        Y
C
      COMMON TIME, R1, DELTMA, SWMAX, R2, DELTMI, DSF, DRF, ICYC, NS,
     1 INTSW
C
      PI = 3.141593
      IOPT = 0
```

```
      IOPEN = 0
      N = 12
   46 CONTINUE
      CALL INPUT
      IF(IPLOT)48,50,48
   48 IF(IOPEN) 50,49,50
   49 CALL PLOTS(BUFF, 12000, 7)
      IOPEN = 1
   50 CONTINUE
      NLOC=16
      K = 0
      IOUT = 3
      LINSPP=50
      LINS=99
      IPAGE=1
      WRITE(IOUT,24)IPAGE
      ICNT = NPNT
      ICNT2 = NPLT
C
C  COMPUTE RHO * L CONSTANTS
C
      RHOH = RHO * .5
      RHOL2 = RHOH * AL * AL
      RHOL3 = RHOL2 * AL
      RHOL4 = RHOL3 * AL
      RHOL5 = RHOL4 * AL
C
C  WRITE OUT HYDRODYNAMIC COEFFICIENTS
C
      WRITE(IOUT,1) XQQ, YRD, ZQD, AKPD, AMQD, ANRD, XRR, YPD, ZPP,
     1 AKRD, AMPP, ANPD, XRP, YPAP, ZRR, AKQR, AMRR, ANPQ, XUD,
     2 YPQ, ZRP, AKPQ, AMRP, ANQR,  XVR, YQR, ZWD, AKPAP, AMQAQ,
     3 ANRAR, XWQ, YVD, ZVR, AKP, AMWD, ANVD
    1 FORMAT(1H ,'XQQ',4X,E12.5,'  YRD',4X,E12.5,'  ZQD',4X,E12.5,
     1'  KPD',4X,F12.5,'  MQD',4X,F12.5,'  NRD',4X,F12.5/1H ,
     2'XRR',4X,E12.5,'  YPD',4X,F12.5,'  ZPP',4X,E12.5,'  KRD',4X,
     3F12.5,'  MPP',4X,F12.5,'  NPD',4X,E12.5/1H ,
     4'XRP',4X,E12.5,'  YPAP',3X,F12.5,'  ZRR',4X,F12.5,'  KQR',4X,
     5F12.5,'  MRR',4X,F12.5,'  NPQ',4X,F12.5/1H ,
     6'XUD',4X,F12.5,'  YPQ', 4X,F12.5,'  ZRP',4X,F12.5,'  KPQ',4X,
     7F12.5,'  MPP',4X,F12.5,'  NQR',4X,F12.5/1H ,
     8'XVR',4X,E12.5,'  YQR',4X,F12.5,'  ZWD',4X,F12.5,'  KPAP',3X,
     9F12.5,'  MQAQ',3X,F12.5,'  NRAR',3X,E12.5/1H ,
     A'XWQ',4X,F12.5,'  YVD',4X,F12.5,'  ZVR',4X,F12.5,'  KP',5X,
     BF12.5,'  MWD',4X,E12.5,'  NVD',4X,F12.5)
      WRITE(IOUT,11)  XUU, YVQ, ZVP, AKR,
     1 AMVR, ANWR, XVV, YWP, ZQ, AKVD, AMVP, ANWP
   11 FORMAT(1H ,'XUU',4X,F12.5,'  YVQ',4X,F12.5,'  ZVP',4X,F12.5,
     1'  KR',5X,F12.5,'  MVR',4X,F12.5,'  NWR',4X,F12.5/1H ,
     2'XVV',4X,E12.5,'  YWP',4X,F12.5,'  ZQ',5X,E12.5,'  KVD',4X,
     3F12.5,'  MVP',4X,E12.5,'  NWP',4X,F12.5)
      WRITE(IOUT,2) XWW, YWR, ZAQDS, AKVQ, AMQ, ANVQ, XDRDR, YR,
     1ZWAQ, AKWP, AMAQDS, ANP, XDSDS, YP, ZSTR, AKWR, AMAWQ, ANR,
     2XDBDB, YARDR, ZW, AKSTR, AMSTR, ANARDR,XVVE,YVAR, ZWAW, AKV,
```

```
3AMW, ANAVR,XWWE,YSTR,ZAW,AKVAV,AMWAW,ANSTR
 2 FORMAT(1H ,'XWW'.4X,E12.5,'  YWR',4X,E12.5,'  ZAQDS',2X,E12.5,
  1'  KVQ',4X,F12.5,'  MQ',5X,E12.5,'  NVQ',4X,F12.5/1H ,
  2'XDRDR',2X,E12.5,'  YR',5X,E12.5,'  ZWAQ',3X,F12.5,'  KWP',4X,
  3E12.5,'  MAQDS',2X,F12.5,'  NP',5X,E12.5/1H ,
  4'XDSDS',2X,E12.5.'  YP',5X,E12.5,'  ZSTR',3X,E12.5,'  KWR',4X,
  5F12.5,'  MAWQ',3X,E12.5,'  NR',5X,E12.5/1H ,
  6'XDBDB',2X,F12.5,'  YARDR',2X,E12.5,'  ZW',5X,E12.5,'  KSTR',3X,
  7F12.5,'  MSTR',3X,F12.5,'  NARDR',2X,E12.5/1H ,
  8'XVVE',3X,F12.5,'  YVAR',3X,E12.5,'  ZWAW',3X,F12.5,'  KV',5X,
  9F12.5,'  MW',5X,E12.5,'  NAVR',3X,E12.5/1H ,
  A'XWWE',3X,E12.5,'  YSTR',3X,E12.5,'  ZAW',4X,E12.5,'  KVAV',3X,
  BE12.5,'  MWAW',3X,E12.5,'  NSTR',3X,F12.5)
   WRITE(IOUT,22)                                  XDRDRE, YV, ZWW,
  1AKVW, AMAW, ANV, XDSDSE, YVAV, ZVV, AKDR, AMWW, ANVAV
22 FORMAT(1H ,'XDRDRE',1X,E12.5,'  YV',5X,E12.5,'  ZWW',4X,F12.5,
  1'  KVW',4X,F12.5,'  MAW',4X,F12.5,'  NV',5X,E12.5/1H ,
  2'XDSDSE',1X,E12.5,'  YVAV',3X,E12.5,'  ZVV',4X,E12.5,'  KDR',4X,
  3F12.5,'  MWW',4X,E12.5,'  NVAV',3X,E12.5)
   WRITE(IOUT,3)      YVW, ZDS, AKSTRE, AMVV, ANVW,          YDR,
  1ZDB, AMDS, ANDR,          YRE, ZQE, AMDB, ANRE, YVE, ZWE, AMQE,
  2ANVE, YVAVE, ZWAWE, AMWE, ANVAVE, YDRE, ZDSE, AMWAWE, ANDRE, AMDSE
 3 FORMAT(1H ,    19X,          '  YVW',4X,E12.5,'  ZDS',4X,E12.5,
  1'  KSTRE',2X,E12.5,'  MVV',4X,F12.5,'  NVW',4X,E12.5/1H ,
  219X,                   '  YDR',4X,F12.5,'  ZDB',4X,F12.5,21X,
  3'  MDS',4X,E12.5,'  NDR',4X,E12.5/1H ,
  419X,                   '  YRE'         ,4X,F12.5,'  ZQE',4X,F12.5,21X,
  5'  MDB',4X,E12.5,'  NRE',4X,E12.5/1H ,
  6 19X,'  YVE',4X,F12.5,'  ZWE',4X,F12.5,21X,'  MQE',4X,E12.5,
  7'  NVE',4X,E12.5/1H , 19X,'  YVAVE',2X,F12.5,'  ZWAWE',2X,E12.5,
  8 21X,'  MWE',4X,E12.5,'  NVAVE',2X,E12.5/1H ,19X,'  YDRE',3X,
  9 E12.5,'  ZDSE',3X,E12.5,21X,'  MWAWE',2X,F12.5,'  NDRE',3X.
  A E12.5,/1H ,84X,'MDSE',3X,F12.5//)
   WRITE(IOUT,4) IX, IY, IZ, IXY, IXZ, IYZ, CW, CB, UC, XB, YB,
  1 ZB, DR, DS, DB, RHO, AL, AM
 4 FORMAT(1H ,'IX',5X,E12.5,'  IY',5X,E12.5,'  IZ',5X,F12.5,
  1'  IXY',4X,E12.5,'  IXZ',4X,F12.5,'  IYZ',4X,F12.5/1H ,
  2'W',6X,E12.5,'  B',6X,E12.5,'  UC',5X,E12.5,'  XB',5X,F12.5,
  3'  YB',5X,F12.5,'  ZB',5X,F12.5/1H ,
  4'DR',5X,F12.5,'  DS',5X,F12.5,'  DB',5X,E12.5,'  RHO',4X,F12.5,
  5'  L',6X,F12.5,'  M'6X,F12.5 )
   WRITE(IOUT,6) A11, A21, A31, DRMAX, ETAHI, ETALO, A12, A22, A32,
  1 XG, YG, ZG, A13, A23, A33
  2, H,INTSW, TIME, R1, DELTMA, SWMAX, R2, DELTMI, DSE, DRE, ICYC, NS
 6 FORMAT(1H ,'A11',4X,F12.5,'  A21',4X,E12.5,'  A31',4X,F12.5,
  1'  DRMAX',2X,E12.5,'  ETAHI',2X,F12.5,'  ETALO',2X,F12.5/
  21H ,'A12',4X,F12.5,'  A22',4X,F12.5,'  A32',4X,F12.5,
  3'  XG',5X,F12.5,'  YG',5X,F12.5,'  ZG',5X,F12.5/
  41H ,'A13',4X,E12.5,'  A23',4X,F12.5,'  A33',4X,F12.5,
  5'  H',4X,E12.5,'  INTSW',2X,I2/
  61H ,'TIME'3X,F12.5,'  R1',5X,F12.5,'  DELTMA',1X,F12.5,'  SWMAX',
  72X,F12.5,'  R2',5X,F12.5,'  DELTMI',1X,F12.5/
  91H ,'DSE',4X,F12.5,'  DRE',4X,F12.5,'  ICYC',3X,I2,10X,
  9'  NS',5X,I2/)
```

```
      WRITE(IOUT,5) IRUN
    5 FORMAT(1H ,'RUN NO  ', I5/)
C
C  COMPUTE W-B
C
      WMB=CW-CB
      IS1 = 0
      ISW2 = 0
      IF(INTSW)60,67,62
   60 INTSW = 1
   62 IF(INTSW-3)66,66,64
   64 INTSW = 1
   66 INTGSW=1
      GO TO 8
   67 INTGSW=0
    8 CONTINUE
      IF(INTGSW)69,68,69
   68 CALL KUTTA(Y)
      GO TO 75
   69 CALL INTEG(Y,INTSW)
   75 CONTINUE
C
C  SAVE VALUES FOR PLOTS IF IPLOT = 1
C
      IF(IPLOT)9,12,9
    9 CONTINUE
C
C IF ARRAY FULL DONT OVERUN
C
      IF(K-300) 63,12,12
   63 CONTINUE
      IF(NPLT-ICNT2) 52,52,51
   52 ICNT2 = 0
      K = K + 1
C  SAVE TIME
      SAVE(K,1) = Y(13)
C  SAVE U
      SAVE(K,2) = Y(1)
C  SAVE V
      SAVE(K,3) = Y(2)
C  SAVE W
      SAVE(K,4) = Y(3)
C  SAVE P
      SAVE(K,5) = Y(4)
C  SAVE Q
      SAVE(K,6) = Y(5)
C  SAVE R
      SAVE(K,7) = Y(6)
C  SAVE THETA
      SAVE(K,8) = Y(7)
C SAVE PSI
      SAVE(K,9) = Y(8)
C  SAVE PHI
      SAVE(K,10) = Y(9)
```

```
C   SAVE X
      SAVE(K,11) = Y(10)
C   SAVE Y
      SAVE(K,12) = Y(11)
C   SAVE Z
      SAVE(K,13) = Y(12)
C   SAVE DR
      SAVE(K,14) = DR
C   SAVE DS
      SAVE(K,15) = DS
C   SAVE DB
      SAVE (K,16) = DB
   51 ICNT2 = ICNT2 + 1
   12 CONTINUE
      IF(NPNT-ICNT) 16,16,17
   16 ICNT = 0
      IF(LINSPP-LINS)20,20,30
   20 LINS=0
      IPAGE=IPAGE+1
      WRITE(IOUT,24)IPAGE
   24 FORMAT(1H1,7X,'EB920',30X,'SUBMARINE SIMULATION',45X,'PAGE',I8//)
      WRITE(IOUT,25)
   25 FORMAT(1H ,9X,'U',16X,'V',16X,'W',16X,'P',16X,'Q',16X,'R',
     1 14X,'THETA'/9X,'PSI',14X,'PHI',15X,'X',16X,'Y',15X,'Z',16X,'T',
     2 16X,'H')
   30 CONTINUE
C
C   IF VARIABLE STEP SIZE NOT BEING USED DONT PRINT H
C
      IF(DH)70,72,70
   70 WRITE(IOUT,10) (Y(I),I=1,13),H
      GO TO 74
   72 WRITE(IOUT,10) (Y(I),I=1,13)
   10 FORMAT(1H0,7(2X,F13.6,2X)/1H ,7(2X,F13.6,2X))
   74 CONTINUE
      LINS=LINS+3
   17 ICNT = ICNT + 1
      IF(ABS(Y(13)-TLIM)-H)35,8,8
   35 CONTINUE
      IF(IPLOT) 40,46,40
   40 CALL PLTROU(SAVE,K,ILOC,NLOC,IRUN)
      GO TO 46
      END
```

```
      SUBROUTINE INPUT
      DIMENSION Y(13), TL(12), ILOC(16), YHOLD(13), COM(219)
      EQUIVALENCE (COM(1),H)
      REAL IX,IY,IZ,IXY,IXZ,IYZ
C
      COMMON H, HMAX, HMIN, DH, FCT, TL, NGS, N, IS1, NPNT
C
      COMMON TLIM, RHOL2, RHOL3, RHOL4, RHOL5, WMB, ETA, ETAM1, ISW2
C
      COMMON          XQQ, XRR, XRP, XUD, XVR ,XWQ, XUU, XVV,
     1                XWW, XDRDR, XDSDS, XDBDB, XVVE, XWWE, XDRDRE, XDSDSE
C
      COMMON          YRD, YPD, YPAP, YPQ, YQR, YVD, YVQ, YWP,
     1                YWR, YR, YP, YARDR, YVAR, YSTR, YV, YVAV,
     2                YVW, YDR, YRE, YVE, YVAVE, YDRE
C
      COMMON          ZQD, ZPP, ZRR, ZRP, ZWD, ZVR, ZVP, ZQ,
     1                ZAQDS, ZWAQ, ZSTR, ZW, ZWAW, ZAW, ZWW, ZVV,
     2                ZDS, ZDB, ZQE, ZWE, ZWAWE, ZDSE
C
      COMMON          AKPD, AKRD, AKQR, AKPQ, AKPAP, AKP, AKR, AKVD,
     1                AKVQ, AKWP, AKWR, AKSTR, AKV, AKVAV, AKVW, AKDR,
     2                AKSTRE
C
      COMMON          AMQD, AMPP, AMRR, AMRP, AMQAQ, AMWD, AMVR, AMVP,
     1                AMQ, AMAQDS, AMAWQ, AMSTR, AMW, AMWAW, AMAW, AMWW,
     2                AMVV, AMDS, AMDB, AMQE, AMWE, AMWAWE, AMDSE
C
      COMMON          ANRD, ANPD, ANPQ, ANQR, ANRAR, ANVD, ANWR, ANWP,
     1                ANVQ, ANP, ANR, ANARDR, ANAVR, ANSTR, ANV, ANVAV,
     2                ANVW, ANDR, ANRE, ANVE, ANVAVE, ANDRE
C
      COMMON           IX, IY, IZ, IXY, IXZ, IYZ
C
      COMMON           CW, CB, UC, XB, YB, ZB
C
      COMMON           DR, DS, DB, RHO, AL, AM
C
      COMMON          DRMAX, ETAHI, ETALO, A11, A12, A13
C
      COMMON          A21, A22, A23, A31, A32, A33
C
      COMMON          XG, YG, ZG
C
      COMMON          ILOC, IPLOT, IRUN, IOPEN, NPLT,IOPT
C
      COMMON          Y
C
      COMMON TIME, R1, DELTMA, SWMAX, R2, DELTMI, DSE, DRE, ICYC, NS,
     1 INTSW
C
      IN  = 1
      IF(IOPT)150,5,150
    5 CONTINUE
```

```
      READ(IN,50) NGS, NPNT, IPLOT, IRUN, NPLT, IOPT, ICYC, NS,INTSW
      IF(IRUN)70,60,70
   60 IF(IOPEN) 62,64,62
   62 CALL PLOT(5.0,0.0,999)
   64 CONTINUE
      CALL EXIT
   70 CONTINUE
      READ(IN,50) (ILCC(I), I = 1, 16)
   50 FORMAT(16I5)
      READ(IN,100)TO,HO,DH,HMAX,HMIN,FCT,TLIM
      H = HO
  100 FORMAT(8F10.5)
      READ(IN,100)(TL(I),I=1,12)
      READ(IN,100)(Y(I), I=1,12)
      Y(13)= TO
C
      READ(IN,100) XQQ, XRR, XRP, XUD, XVR ,XWQ, XUU, XVV,
     1XWW, XDRDR, XDSDS, XDBDB, XVVE, XWWE, XDRDRE, XDSDSE
C
      READ(IN,100) YRD, YPD, YPAP, YPQ, YQR, YVD, YVQ, YWP,
     1             YWR, YR, YP, YARDR, YVAR, YSTR, YV, YVAV,
     2             YVW, YDR, YRE, YVE, YVAVE, YDRE
C
      READ(IN,100) ZQD, ZPP, ZRR, ZRP, ZWD, ZVR, ZVP, ZQ,
     1             ZAQDS, ZWAQ, ZSTR, ZW, ZWAW, ZAW, ZWW, ZVV,
     2             ZDS, ZDB, ZQE, ZWE, ZWAWE, ZDSE
C
      READ(IN,100) AKPD, AKRD, AKQR, AKPQ, AKPAP, AKP, AKR, AKVD,
     1             AKVQ, AKWP, AKWR, AKSTR, AKV, AKVAV, AKVW, AKDR,
     2             AKSTRE
C
      READ(IN,100) AMQD, AMPP, AMRR, AMRP, AMQAQ, AMWD, AMVR, AMVP,
     1             AMQ, AMAQDS, AMAWQ, AMSTR, AMW, AMWAW, AMAW, AMWW,
     2             AMVV, AMDS, AMDB, AMQE, AMWE, AMWAWE, AMDSE
C
      READ(IN,100) ANRD, ANPD, ANPQ, ANQR, ANRAR, ANVD, ANWR, ANWP,
     1             ANVQ, ANP, ANR, ANARDR, ANAVR, ANSTR, ANV, ANVAV,
     2             ANVW, ANDR, ANRE, ANVE, ANVAVE, ANDRE
C
      READ(IN,100)  IX, IY, IZ, IXY, IXZ, IYZ
C
      READ(IN,100)  CW, CB, UC, XB, YB, ZB
C
      READ(IN,100)  DR, DS, DB, RHO, AL, AM
C
      READ (IN,100) DRMAX, ETAHI, ETALO, A11, A12, A13
C
      READ(IN,100)  A21, A22, A23, A31, A32, A33
C
      READ(IN,100) XG, YG, ZG
C
      READ(IN,100) TIME, PI, DELTMA, SWMAX, R2, DELTMI, DSE, DRE
C
C  SAVE  INITIAL  CONDITIONS  FOR  POSSIBLE  RESTORE
```

```
C
      DO 110 I=1, 13
      YHOLD(I)= Y(I)
  110 CONTINUE
C
C  SAVE  DR,DS,DB  FOR  POSSIBLE  RESTORE
C
      DRHOLD = DR
      DSHOLD = DS
      DBHOLD = DB
      RETURN
  150 CONTINUE
C
C  RESTORE  INITIAL VALUES
C
      DO 152 I= 1,13
      Y(I) = YHOLD(I)
  152 CONTINUE
C
C  RESTORE  INITIAL  DR,DS,DB,H
C
      DR  = DRHOLD
      DS  =  DSHOLD
      DB  = DBHOLD
      H   =  H0
      READ(IN,165) IRUN
      IF(IRUN) 155,05,155
  155 CONTINUE
  160 READ(IN,165) NDEX, VALUE
  165 FORMAT(I5,5X,F10.5)
      IF(NDEX)180,170,180
  170 RETURN
  180 COM(NDEX) = VALUE
      GO TO 160
      END
```

```
      SUBROUTINE KUTTA(Y)
      DIMENSION Y(1), F(12), Y2(13), Q(12), TL(12)
      COMMON H, HMAX, HMIN, DH, FCT, TL, NGS, N, IS1
      IF(IS1)2,1,2
    1 M = N + 1
      K = 0
      IS1 = 1
      DH4 = DH* DH * DH * DH
      CALL EVAL (Y,F)
      RETURN
    2 DO 3 I = 1,N
      Y2(I) = Y(I) + .5*H*F(I)
    3 Q(I) = F(I)
      Y2 (M) = Y(M) + .5*H
      CALL EVAL(Y2,F)
      DO 4 I = 1,N
      Y2(I) = Y(I) + .5*H*F(I)
    4 Q(I) = Q(I) + F(I) + F(I)
      CALL EVAL(Y2,F)
      DO 5 I = 1,N
      Y2(I) = Y(I) + H * F(I)
    5 Q(I) = Q(I) + F(I) + F(I)
      Y2(M) = Y(M) + H
      CALL EVAL (Y2,F)
      DO 6 I = 1,N
      Y2(I) = Y(I) + .16666667 * H * (Q(I) + F(I))
    6 Q(I) = F(I)
      CALL EVAL (Y2,F)
      IF(DH)7,13,7
    7 DO 8 I = 1,N
      Q(I) = H * ABS(F(I) - Q(I))
      IF(Q(I) - TL(I))8,8,15
    8 CONTINUE
      DO 9 I = 1,N
      Q(I) = FCT * DH4 * Q(I)
      IF(Q(I) - TL(I))9,9,12
    9 CONTINUE
      K = K + 1
      IF (K - NGS)13,10,10
   10 IF(H - HMAX)11,12,12
   11 H = H * DH
   12  K = 0
   13 DO 14 I = 1,N
   14 Y(I) = Y2(I)
      Y(M) = Y2(M)
      RETURN
   15 IF(H-HMIN)12,12,16
   16 H = H/DH
      CALL EVAL(Y,F)
      GO TO 2
      END
```

```
      SUBROUTINE EVAL(YI,F)
      DIMENSION YI(1),F(1),TL(12),A(6,6),B(6)
      REAL IX,IY,IZ,IXY,IXZ,IYZ
C
      COMMON H, HMAX, HMIN, DH, FCT, TL, NGS, N, IS1, NPNT
C
      COMMON TLIM, RHOL2, RHOL3, RHOL4, RHOL5, WMB, ETA, ETAM1, ISW2
C
      COMMON         XQQ, XRR, XRP, XUD, XVR ,XWQ, XUU, XVV,
     1               XWW, XDRDR, XDSDS, XDBDB, XVVE, XWWE, XDRDRE, XDSDSE
C
      COMMON         YRD, YPD, YPAP, YPQ, YQR, YVD, YVQ, YWP,
     1               YWR, YR, YP, YARDP, YVAR, YSTR, YV, YVAV,
     2               YVW, YDR, YRE, YVE, YVAVE, YDRE
C
      COMMON         ZQD, ZPP, ZRR, ZRP, ZWD, ZVR, ZVP, ZQ,
     1               ZAQDS, ZWAQ, ZSTP, ZW, ZWAW, ZAW, ZWW, ZVV,
     2               ZDS, ZDB, ZQE, ZWE, ZWAWE, ZDSE
C
      COMMON         AKPD, AKRD, AKQR, AKPQ, AKPAP, AKP, AKR, AKVD,
     1               AKVQ, AKWP, AKWR, AKSTR, AKV, AKVAV, AKVW, AKDR,
     2               AKSTRE
C
      COMMON         AMQD, AMPP, AMRR, AMRP, AMQAQ, AMWD, AMVR, AMVP,
     1               AMQ, AMAQDS, AMAWQ, AMSTR, AMW, AMWAW, AMAW, AMWW,
     2               AMVV, AMDS, AMDB, AMQE, AMWE, AMWAWE, AMDSE
C
      COMMON         ANRD, ANPD, ANPQ, ANQR, ANRAR, ANVD, ANWR, ANWP,
     1               ANVQ, ANP, ANR, ANARDR, ANAVR, ANSTR, ANV, ANVAV,
     2               ANVW, ANDR, ANRE, ANVE, ANVAVE, ANDRE
C
      COMMON          IX, IY, IZ, IXY, IXZ, IYZ
C
      COMMON          CW, CB, UC, XB, YB, ZB
C
      COMMON          DR, DS, DB, RHO, AL, AM
C
      COMMON         DRMAX, ETAHI, ETALO, A11, A12, A13
C
      COMMON         A21, A22, A23, A31, A32, A33
C
      COMMON         XG, YG, ZG
C
      EQUIVALENCE(B(1),FA),(B(2),FL),(B(3),FN),(B(4),RM),(B(5),PM),
     1 (B(6),YM)
C
C  PULL PRESENT VALUES OF VARIABLES OUT OF ARRAY YI
C
      U = YI(1)
      V = YI(2)
      W = YI(3)
      P = YI(4)
      Q = YI(5)
      R = YI(6)
```

```
      THETA =YI(7)
      PSI = YI(8)
      PHI = YI(9)
      X = YI(10)
      Y = YI(11)
      Z = YI(12)
      CALL CONTR(THETA)
C
C COMPUTE QUANTITIES TO BE USED MORE THAN ONCE
C
      VR = V*R
      Q2 = Q*Q
      R2 = R*R
      RP = R*P
      U2 = U*U
      V2 = V*V
      W2 = W*W
      DR2 = DR*DR
      DR2U2 = DR2 * U2
      DS2 = DS*DS
      DS2U2 = DS2*U2
      WP = W*P
      UR = U*R
      PABSP = P*ABS(P)
      PQ = P*Q
      QR = Q*R
      VQ = V*Q
      WQ = W*Q
      WR = W*R
      UP = U*P
      ROOTVW = SQRT(V2+W2)
      VRTVW = V*ROOTVW
      ABSR = ABS(R)
      UARDR=U*ABSR*DR
      UV=U*V
      VW=V*W
      DRU2=DR*U2
      UQ=U*Q
      VP=V*P
      P2=P*P
      ABSQ=ABS(Q)
      UAQDS=U*ABSQ*DS
      WRTVW=W*ROOTVW
      UW=U*W
      ABSW=ABS(W)
      UABSW=U*ABSW
      CBU2=DB*U2
      DSU2=DS*U2
      ABSV = ABS(V)
      UMAG = SQRT(U2+V2+W2)
      IF(UMAG)26,24,26
   24 ETA = 20.
      GO TO 28
   26 CONTINUE
```

```
      ETA = UC/UMAG
   28 ETAM1 = ETA-1.
      IF(ETA-ETAHI)32,30,30
   30 A1=A11
      A2=A12
      A3=A13
      GO TO 38
   32 IF(ETA-ETALO)35,35,37
   35 A1=A31
      A2=A32
      A3=A33
      GO TO 38
   37 A1=A21
      A2=A22
      A3=A23
   38 CONTINUE
      IF(V) 2,1,2
    1 RATVAV = 0.
      GO TO 3
    2 RATVAV = V/ABSV
    3 IF(W) 5,4,5
    4 RATWAW = 0.
      GO TO 6
    5 RATWAW = W/ABSW
    6 CONTINUE
C
C   COMPUTE TRIG FUNCTIONS
C
      SPHI = SIN(PHI)
      CPHI = COS(PHI)
      STTA = SIN(THETA)
      CTTA = COS(THETA)
      SPSI = SIN(PSI)
      CPSI = COS(PSI)
      TRIG1 = CTTA*SPHI
      TRIG2 = CTTA*CPHI
      TRIG3=SPHI*STTA
      TRIG4 = CPHI * STTA
      TRIG5 = U*CTTA
      IF(ISW2)20,10,20
   10 ISW2 = 1
C
C   SET COEFFICIENTS OF UD,VD,WD,PD,QD,RD, IN MATRIX FOR INVERTING
C
      A(1,1)= AM-RHOL3*XUD
      A(1,2)= 0.
      A(1,3)= 0.
      A(1,4)= 0.
      A(1,5)= AM*ZG
      A(1,6)= -AM*YG
      A(2,1) = 0.
      A(2,2) = AM-RHOL3*YVD
      A(2,3) = 0.
      A(2,4) = -RHOL4*YPD-AM*ZG
```

```
      A(2,5) = 0.
      A(2,6) = -RHOL4*YRD+AM*XG
      A(3,1) = 0.
      A(3,2) = 0.
      A(3,3) = AM-RHOL3*ZWD
      A(3,4) = AM*YG
      A(3,5) = -RHOL4*ZQD-AM*XG
      A(3,6) = 0.
      A(4,1) = 0.
      A(4,2) = -RHOL4*AKVD-AM*ZG
      A(4,3) = AM*YG
      A(4,4) = IX-RHOL5*AKPD
      A(4,5) = -IXY
      A(4,6) = -IXZ-RHOL5*AKRD
      A(5,1) = AM*ZG
      A(5,2) = 0.
      A(5,3) = -RHOL4*AMWD-AM*XG
      A(5,4) = -IXY
      A(5,5) = IY-RHOL5*AMQD
      A(5,6) = -IYZ
      A(6,1) = -AM*YG
      A(6,2) = -RHOL4*ANVD+AM*XG
      A(6,3) = 0.
      A(6,4) = -IXZ-RHOL5*ANPD
      A(6,5) = -IYZ
      A(6,6) = IZ-RHOL5*ANRD
C
C  INVERT A MATRIX
C
      EPS =.000001
      CALL INVER2(A,6,6,6,IER,EPS)
      IF(IER)14,16,14
   14 WRITE(3,15)
   15 FORMAT(1H ,'SINGULAR MATRIX')
      CALL EXITD
   16 CONTINUE
C
   20 CONTINUE
C
C  COMPUTE RIGHT SIDE OF AXIAL FORCE EQN
C
   40 CONTINUE
      FA=  (AM*(VR-WQ) + RHOL4*(XQQ*Q2+XRR*R2+XRP*RP) +
     1RHOL3*(          XVR*VR+XWQ*WQ)+RHOL2*(XUU*U2+XVV*V2+XWW*W*W)+
     2RHOL2*U2*(XDRDR*DR2   +XDSDS*DS2+XDBDB*DB*DB) +
     3RHOL2*(A1*U2+A2*U*UC+A3*UC*UC)-WMB*STTA+ RHOL2*(XVVE*V2+XWWE*W2+
     4XDRDRE*DR2U2 + XDSDSE*DS2U2)*ETAM1)
     5 +AM*(XG*(Q2+R2)-YG*PQ-ZG*RP)
C
C  COMPUTE RIGHT SIDE OF LATERAL FORCE EQN
C
   50 CONTINUE
      FL = AM*(WP-UR)+RHOL4*(YPAP*PABSP+YPQ*PQ+YQR*QR) +
     1RHOL3*(YVQ*VQ+YWP*WP+YWR*WR+YR*UR+YP*UP+YARDR*U*ABSR*DR  +
```

```
      2YVAR*RATVAV*ROOTVW*ABSR)+RHOL2*(YSTR*U2+YV*UV+YVAV*VRTVW +
      3YVW*VW+YDR*DRU2) + WMB*TRIG1+RHOL3*YRE*UR*ETAM1+
      4RHOL2*(YVE*UV+YVAVE*VRTVW+YDRE*DRU2)*ETAM1
      5 +AM*(YG*(R2+P2)-ZG*QR-XG*PQ)
C
C   COMPUTE RIGHT SIDE OF NORMAL FORCE EQN
C
   60 CONTINUE
       FN = AM*(UQ-VP) + RHOL4 * (ZPP*P2+ZRR*R2+ZRP*RP) + RHOL3*(ZVR*VR+
      1ZVP*VP +ZQ*UQ +ZAQDS* UAQDS + ZWAQ* RATWAW*ROOTVW*ABSQ) +
      2RHOL2 *(ZSTR*U2 + ZW*UW+ZWAW * WRTVW +ZAW*UABSW+ZWW*ABSW*ROOTVW
      3+ ZVV*V2+ZDS*DSU2 +ZDB*DBU2) + WMB * TRIG2 + RHOL3* ZQE*UQ* ETAM1
      4+RHOL2*(ZWE*UW+ZWAWE*WRTVW+ ZDSE*DSU2)*ETAM1
      5 +AM*(ZG*(P2+Q2)-XG*RP-YG*QR)
C
C   COMPUTE RIGHT SIDE OF ROLLING MOMENT EQN
C
   70 CONTINUE
       RM=(IY-IZ)*QR+IXZ*PQ+IYZ*(Q2-R2)-IXY*RP+RHOL5*(AKQR*QR +
      1AKPQ*PQ+AKPAP*PABSP)+RHOL4*(AKP*UP+AKR*UR+AKVQ*VQ+AKWP*WP
      2+AKWR*WR)+RHOL3*(AKSTR*U2+AKV*UV+AKVAV*VRTVW+AKVW*VW+AKDR*DRU2)
      3+(YG*CW-YB*CB)*TRIG2-(ZG*CW-ZB*CB)*TRIG1+RHOL3*AKSTRE*U2*ETAM1
      4 +AM*(YG*(UQ-VP)+ZG*(UR-WP))
C
C   COMPUTE RIGHT SIDE OF PITCHING MOMENT EQN
C
   80 CONTINUE
       PM = (IZ-IX)*RP+IXY*QR+(R2-P2)*IXZ-IYZ*PQ +RHOL5*(AMPP*P2+AMRR*
      1R2 +AMRP*RP +AMQAQ*Q*ABSQ)+RHOL4*(AMVR*VR+AMVP*VP+AMQ*UQ+AMAQDS*
      2UAQDS +AMAWQ *Q*ROOTVW)+ RHOL3*(AMSTR*U2+AMW*UW+AMWAW*WRTVW +
      3AMAW* UABSW +AMWW*ABSW*ROOTVW+AMVV*V2+AMDS*DSU2+AMDB*DBU2)
      4-(XG*CW-XB*CB)*TRIG2-(ZG*CW-ZB*CB)*STTA+RHOL4*AMQE*UQ*ETAM1
      5+RHOL3*(AMWE*UW+AMWAWE*WRTVW+AMDSE*DSU2)*ETAM1
      6 +AM*(ZG*(VR-WQ)+XG*(VP-UQ))
C
C   COMPUTE RIGHT SIDE OF YAWING MOMENT EQN
C
   90 CONTINUE
       YM = (IX-IY)*PQ +IYZ*RP +(P2-Q2)*IXY - IXZ*QR +  RHOL5*(ANPQ*
      1PQ + ANQR*QR +ANRAR*R*ABSR)+RHOL4*(ANWR*WR+ANWP*WP +ANVQ*VQ +
      2ANP*UP+ ANR*UR + ANARDR* UARDR + ANAVR * R*ROOTVW)  +
      3RHOL3*(ANSTR*U2 +ANV*UV+ANVAV*VRTVW+ANVW*VW+ANDR*DRU2)
      4+(XG*CW-XB*CB)*TRIG1+(YG*CW-YB*CB)*STTA+RHOL4*ANRE*UR*ETAM1 +
      5RHOL3*(ANVE*UV+ANVAVE*VRTVW +ANDRE*DRU2)*ETAM1
      6 +AM*(XG*(WP-UR)+YG*(WQ-VR))
C
C   MULTIPLY TO GET UD, VD, WD, PD, QD, RD
C
       CALL MATMPY(A,B,F,6,6,1,6,6,6,1)
C
C   COMPUTE KINEMATICS - THETA DOT , PSI DOT, PHI DOT
C
       F(7) = Q*CPHI-R*SPHI
       F(8) =(Q*SPHI+R*CPHI)/CTTA
```

```
      F(9) =P+F(8)*STTA
C
C  COMPUTE  X DOT , Y DOT , Z DOT
C
      F(10)=TRIG5*CPSI+V*(TRIG3*CPSI-CPHI*SPSI)+
     1 W*(TRIG4*CPSI + SPHI*SPSI)
      F(11)=TRIG5*SPSI+V*(TRIG3*SPSI+CPHI*CPSI) +
     1 W*(TRIG4*SPSI-SPHI*CPSI)
      F(12)=-U*STTA+V*TRIG1+W*TRIG2
      RETURN
      END
```

```
      SUBROUTINE INVER2(A,M,N,L,IC,EPS)
C     A THE INPUT MATRIX CONTAINS THE INVERTED  MATRIX A-1 UPON EXIT
C     THE SOLUTION IS STORED IN A (I,J) , M<J<N
C     M IS THE NUMBER OF ROWS STORED IN MATRIX A.
C     N IS THE NUMBER OF COLUMNS STORED IN MATRIX A.
C     L IS THE MAX NUMBER OF ROWS ALLOCATED IN A.
C     L IS VALUE XX  IN  DIMENSION  A(XX,YY)
C     IC IS 0 IF A IS INVERTED SUCESSFULLY, IF EACH DIAGONAL ELEMENT
C                    IS GREATER IN ABS THAN EPS.
C     IC IS 1 IF A IS NOT INVERTED SUCESSFULLY.
C     EPS IS A VALUE SAY .00001 THAT IS USED FOR SINGULARITY CHECKING.
C     FOR A DOUBLE PRECISION VERSION CHANGE ABS TO DABS IN 5, AND THE
C     LITERAL 1. IN STATEMENT 10 TO 1.D0
      DIMENSION A(1)
C     DOUBLE PRECISION A
      IC = 0
      DO 80 I=1,M
      LI = L*I-L
      II = LI+I
 5    IF(ABS(A(II))-EPS)90,90,10
 10   A(II) = 1./A(II)
      DO  50 K = 1,N
      IF(K-I)20,50,20
 20   LK = L*K-L
      IK = LK+I
      A(IK) = A(IK)* A(II)
      DO 40 J=1,M
      IF(J-I)30,40,30
 30   JI = LI+J
      JK = LK+J
      A(JK) = A(JK) - A(JI)*A(IK)
 40   CONTINUE
 50   CONTINUE
      DO 70 J=1,M
      IF(J-I)60,70,60
 60   JI = LI+J
      A(JI) = - A(JI)* A(II)
 70   CONTINUE
 80   CONTINUE
      RETURN
 90   IC = 1
      RETURN
      END
```

```
      SUBROUTINE MATMPY (A,B,C,M,N,L,MA,MB,MC,IOPT)
      DIMENSION A(1),B(1),C(1)
C     DOUBLE PRECISION A,B,C
      KK= -MB
      II = -MC
      LLL = 0
   30 LLL=LLL+1
      II=II+MC
      I = II
      KK= KK+MB
      III=0
      JJ= -MA+1
   40 K=KK
      J=JJ
      I=I+1
      KKK=0
      III=III+1
      C(I)=0.
   50 K=K+1
      J=MA+J
      KKK=KKK+1
      C(I)=C(I)+ A(J)*B(K)
      IF(KKK-N) 50,60,60
   60 JJ=JJ+1
      IF(III-M)40,70,70
   70 IF(LLL-L)30,80,80
   80 RETURN
      END
```

```
      SUBROUTINE INTEG(Y,INTSW)
      DIMENSION Y(1), F(12), TL(12), F1(12)
      COMMON H, HMAX, HMIN, DH, FCT, TL, NGS, N, IS1
      IF(IS1)30,10,30
   10 M= N + 1
      DO 20 I = 1,N
      F(I) = 0.
   20 F1(I) = 0.
      IS1 = 1
      RETURN
   30 CALL FVAL1(Y,F)
      DO 40 I = 1,N
      GO TO (42,44,46),INTSW
   42 CONTINUE
C
      Y(I) = Y(I) + .5*H*(3.*F(I)-F1(I))
C
      GO TO 48
   44 CONTINUE
C
      Y(I) = Y(I) + .25*H*(3.*F(I)+F1(I))
C
      GO TO 48
   46 CONTINUE
C
      Y(I) = Y(I) + H * F(I)
C
   48 CONTINUE
      F1(I) = F(I)
   40 CONTINUE
      Y(M) = Y(M)+H
      RETURN
      END
```

```
      SUBROUTINE EVAL1(YI,F)
      DIMENSION YI(1),F(1),TL(12)
      REAL IX,IY,IZ,IXY,IXZ,IYZ
C
      COMMON H, HMAX, HMIN, DH, FCT, TL, NGS, N, IS1, NPNT
C
      COMMON TLIM, RHOL2, RHOL3, RHOL4, RHOL5, WMB, ETA, ETAM1, ISW2
C
      COMMON        XQQ, XRR, XRP, XUD, XVR ,XWQ, XUU, XVV,
     1              XWW, XDRDR, XDSDS, XDBDB, XVVE, XWWE, XDRDRE, XDSDSE
C
      COMMON        YRD, YPD, YPAP, YPQ, YQR, YVD, YVQ, YWP,
     1              YWR, YR, YP, YARDR, YVAR, YSTR, YV, YVAV,
     2              YVW, YDR, YRE, YVE, YVAVE, YDRE
C
      COMMON        ZQD, ZPP, ZRR, ZRP, ZWD, ZVR, ZVP, ZQ,
     1              ZAQDS, ZWAQ, ZSTR, ZW, ZWAW, ZAW, ZWW, ZVV,
     2              ZDS, ZDB, ZQE, ZWE, ZWAWE, ZDSE
C
      COMMON        AKPD, AKRD, AKQR, AKPQ, AKPAP, AKP, AKR, AKVD,
     1              AKVQ, AKWP, AKWR, AKSTR, AKV, AKVAV, AKVW, AKDR,
     2              AKSTRE
C
      COMMON        AMQD, AMPP, AMRR, AMRP, AMQAQ, AMWD, AMVR, AMVP,
     1              AMQ, AMAQDS, AMAWQ, AMSTR, AMW, AMWAW, AMAW, AMWW,
     2              AMVV, AMDS, AMDB, AMQE, AMWE, AMWAWE, AMDSE
C
      COMMON        ANRD, ANPD, ANPQ, ANQR, ANRAR, ANVD, ANWR, ANWP,
     1              ANVQ, ANP, ANR, ANARDR, ANAVR, ANSTR, ANV, ANVAV,
     2              ANVW, ANDR, ANRE, ANVE, ANVAVE, ANDRE
C
      COMMON         IX, IY, IZ, IXY, IXZ, IYZ
C
      COMMON         CW, CB, UC, XB, YB, ZB
C
      COMMON         DR, DS, DB, RHO, AL, AM
C
      COMMON        DRMAX, ETAHI, ETALO, A11, A12, A13
C
      COMMON        A21, A22, A23, A31, A32, A33
C
      COMMON        XG, YG, ZG
C
C
C   PULL PRESENT VALUES OF VARIABLES OUT OF ARRAY YI
C
      U = YI(1)
      V = YI(2)
      W = YI(3)
      P = YI(4)
      Q = YI(5)
      R = YI(6)
      THETA =YI(7)
      PSI = YI(8)
```

```
      PHI = YI(9)
      X = YI(10)
      Y = YI(11)
      Z = YI(12)
      CALL CONTR(THETA)
C
C COMPUTE QUANTITIES TO BE USED MORE THAN ONCE
C
      VR = V*R
      Q2 = Q*Q
      R2 = R*R
      RP = R*P
      U2 = U*U
      V2 = V*V
      W2 = W*W
      DR2 = DR*DR
      DR2U2 = DR2 * U2
      DS2 = DS*DS
      DS2U2 = DS2*U2
      WP = W*P
      UR = U*R
      PABSP = P*ABS(P)
      PQ = P*Q
      QR = Q*R
      VQ = V*Q
      WQ = W*Q
      WR = W*R
      UP = U*P
      ROOTVW = SQRT(V2+W2)
      VRTVW = V*ROOTVW
      ABSR = ABS(R)
      UARDR=U*ABSR*DR
      UV=U*V
      VW=V*W
      DRU2=DR*U2
      UQ=U*Q
      VP=V*P
      P2=P*P
      ABSQ=ABS(Q)
      UAQDS=U*ABSQ*DS
      WRTVW=W*ROOTVW
      UW=U*W
      ABSW=ABS(W)
      UABSW=U*ABSW
      DBU2=DR*U2
      DSU2=DS*U2
      ABSV = ABS(V)
      UMAG = SQRT(U2+V2+W2)
      IF(UMAG)26,24,26
   24 ETA = 20.
      GO TO 28
   26 CONTINUE
      ETA = UC/UMAG
   28 ETAM1 = ETA-1.
```

```
      ETAM1 = ETA-1.
      IF(ETA-ETAHI)32,30,30
   30 A1=A11
      A2=A12
      A3=A13
      GO TO 38
   32 IF(ETA-ETALO)35,35,37
   35 A1=A31
      A2=A32
      A3=A33
      GO TO 38
   37 A1=A21
      A2=A22
      A3=A23
   38 CONTINUE
      IF(V) 2,1,2
    1 RATVAV = 0.
      GO TO 3
    2 RATVAV = V/ABSV
    3 IF(W) 5,4,5
    4 RATWAW = 0.
      GO TO 6
    5 RATWAW = W/ABSW
    6 CONTINUE
C
C   COMPUTE TRIG FUNCTIONS
C
      SPHI = SIN(PHI)
      CPHI = COS(PHI)
      STTA = SIN(THETA)
      CTTA = COS(THETA)
      SPSI = SIN(PSI)
      CPSI = COS(PSI)
      TRIG1 = CTTA*SPHI
      TRIG2 = CTTA*CPHI
      TRIG3=SPHI*STTA
      TRIG4 = CPHI * STTA
      TRIG5 = U*CTTA
      IF(ISW2)20,10,20
   10 ISW2 = 1
C
C   SET COEFFICIENTS OF UD,VD,WD,PD,QD,RD
C
      FAU     = AM-RHOL3*XUD
      FAQ     = AM*ZG
      FAR     = -AM*YG
      FLV      = AM-RHOL3*YVD
      FLP      = -RHOL4*YPD-AM*ZG
      FLR      = -RHOL4*YRD+AM*XG
      FNW      = AM-RHOL3*ZWD
      FNP      = AM*YG
      FNQ      = -RHOL4*ZQD-AM*XG
      RMV      = -RHOL4*AKVD-AM*ZG
      RMW      = AM*YG
```

```
      RMP      =  IX-RHOL5*AKPD
      RMQ      =  -IXY
      RMR      =  -IXZ-RHOL5*AKRD
      PMU      =  AM*ZG
      PMW      =  -RHOL4*AMWD-AM*XG
      PMP      =  -IXY
      PMQ      =  IY-RHOL5*AMQD
      PMR      =  -IYZ
      YMU      =  -AM*YG
      YMV      =  -RHOL4*ANVD+AM*XG
      YMP      =  -IXZ-RHOL5*ANPD
      YMQ      =  -IYZ
      YMR      =  IZ-RHOL5*ANRD
C
   20 CONTINUE
C
C  COMPUTE UD FROM AXIAL FORCE EQN
C
   40 CONTINUE
      F(1)=((AM*(VR-WQ) + RHOL4*(XQQ*Q2+XRR*R2+XRP*RP) +
     1RHOL3*(          XVR*VR+XWQ*WQ)+RHOL2*(XUU*U2+XVV*V2+XWW*W*W)+
     2RHOL2*U2*(XDRDR*DR2   +XDSDS*DS2+XDBDB*DB*DB) +
     3RHOL2*(A1*U2+A2*U*UC+A3*UC*UC)-WMB*STTA+ RHOL2*(XVVE*V2+XWWE*W2+
     4XDRDRE*DR2U2 + XDSDSE*DS2U2)*ETAM1)
     5 +AM*(XG*(Q2+R2)-YG*PQ-ZG*RP)
     6 -FAQ* F(5)-FAR* F(6))/FAU
C
C  COMPUTE VD FROM LATERAL FORCE EQN
C
   50 CONTINUE
      F(2)=(AM*(WP-UR)+RHOL4*(YPAP*PABSP+YPQ*PQ+YQR*QR) +
     1RHOL3*(YVQ*VQ+YWP*WP+YWR*WR+YR*UR+YP*UP+YARDR*U*ABSR*DR +
     2YVAR*RATVAV*ROOTVW*ABSR)+RHOL2*(YSTR*U2+YV*UV+YVAV*VRTVW +
     3YVW*VW+YDR*DRU2) + WMB*TRIG1+RHOL3*YRE*UR*ETAM1+
     4RHOL2*(YVE*UV+YVAVE*VRTVW+YDRE*DRU2)*ETAM1
     5 +AM*(YG*(R2+P2)-ZG*QR-XG*PQ)
     6 -FLP* F(4)-FLR*F(6))/FLV
C
C  COMPUTE WD FROM NORMAL FORCE EQN
C
   60 CONTINUE
      F(3)=(AM*(UQ-VP) + RHOL4 * (ZPP*P2+ZRR*R2+ZRP*RP) + RHOL3*(ZVR*VR+
     1ZVP*VP +ZQ*UQ +ZAQDS* UAQDS + ZWAQ* RATWAW*ROOTVW*ABSQ) +
     2RHOL2 *(ZSTP*U2 + ZW*UW+ZWAW * WRTVW +ZAW*UABSW+ZWW*APSW*ROOTVW
     3+ ZVV*V2+ZDS*DSU2 +ZDB*DBU2) + WMB * TRIG2 + RHOL3* ZQE*UQ* ETAM1
     4+RHOL2*(ZWE*UW+ZWAWE*WRTVW+ZDSE*DSU2)*ETAM1
     5 +AM*(ZG*(P2+Q2)-XG*RP-YG*QR)
     6 -FNP* F(4)-FNQ*F(5))/FNW
C
C  COMPUTE PD FROM ROLLING MOMENT EQN
C
   70 CONTINUE
      F(4)=((IY-IZ)*QR+IXZ*PQ+IYZ*(Q2-R2)-IXY*RP+RHOL5*(AKQR*QR +
     1AKPQ*PQ+AKPAP*PABSP)+RHOL4*(AKP*UP+AKR*UR+AKVQ*VQ+AKWP*WP
```

```
      2+AKWR*WR)+RHOL3*(AKSTR*U2+AKV*UV+AKVAV*VRTVW+AKVW*VW+AKDR*DRU2)
      3+(YG*CW-YB*CB)*TRIG2-(ZG*CW-ZB*CB)*TRIG1+RHOL3*AKSTRE*U2*ETAM1
      4 +AM*(YG*(UQ-VP)+ZG*(UR-WP))
      5 -RMV*F(2)-RMW*F(3)-RMQ*F(5)-RMR*F(6))/RMP
C
C     COMPUTE QD FROM PITCHING MOMENT EQN
C
   80 CONTINUE
       F(5)=((IZ-IX)*RP+IXY*QR+(R2-P2)*IXZ-IYZ*PQ  +RHOL5*(AMPP*P2+AMRR*
      1R2  +AMRP*RP  +AMQAQ*Q*ABSQ)+RHOL4*(AMVR*VR+AMVP*VP+AMQ*UQ+AMAQDS*
      2UAQDS  +AMAWQ  *Q*ROOTVW)+ RHOL3*(AMSTR*U2+AMW*UW+AMWAW*WRTVW +
      3AMAW*  UABSW  +AMWW*ABSW*ROOTVW+AMVV*V2+AMDS*DSU2+AMDB*DBU2)
      4-(XG*CW-XB*CB)*TRIG2-(ZG*CW-ZB*CB)*STTA+RHOL4*AMQE*UQ*ETAM1
      5+RHOL3*(AMWE*UW+AMWAWE*WRTVW+AMDSE*DSU2)*ETAM1
      6 +AM*(ZG*(VR-WQ)+XG*(VP-UQ))
      7 -PMU*F(1)-PMW*F(3)-PMP*F(4)-PMR*F(6))/PMQ
C
C     COMPUTE RD FROM YAWING MOMENT EQN
C
   90 CONTINUE
       F(6)=((IX-IY)*PQ +IYZ*RP +(P2-Q2)*IXY  -  IXZ*QR  +   RHOL5*(ANPQ*
      1PQ  +  ANQR*QR  +ANRAR*R*ABSR)+RHOL4*(ANWR*WR+ANWP*WP  +ANVQ*VQ +
      2ANP*UP+  ANR*UR  +  ANARDR*  UARDR  +  ANAVR  *  R*ROOTVW)   +
      3RHOL3*(ANSTR*U2  +ANV*UV+ANVAV*VRTVW+ANVW*VW+ANDR*DRU2)
      4+(XG*CW-XB*CB)*TRIG1+(YG*CW-YB*CB)*STTA+RHOL4*ANRE*UR*ETAM1  +
      5RHOL3*(ANVE*UV+ANVAVE*VRTVW  +ANDRE*DRU2)*ETAM1
      6  +AM*(XG*(WP-UR)+YG*(WQ-VR))
      7  -YMU*F(1)-YMV*F(2)-YMP*F(4)-YMQ*F(5))/YMR
C
C     COMPUTE KINEMATICS - THETA DOT , PSI DOT, PHI DOT
C
       F(7)  =  Q*CPHI-R*SPHI
       F(8)  =(Q*SPHI+R*CPHI)/CTTA
       F(9)  =P+F(8)*STTA
C
C     COMPUTE   X DOT  ,  Y DOT  ,  Z DOT
C
       F(10)=TRIG5*CPSI+V*(TRIG3*CPSI-CPHI*SPSI)+
      1 W*(TRIG4*CPSI  +  SPHI*SPSI)
       F(11)=TRIG5*SPSI+V*(TRIG3*SPSI+CPHI*CPSI)  +
      1 W*(TRIG4*SPSI-SPHI*CPSI)
       F(12)=-U*STTA+V*TRIG1+W*TRIG2
       RETURN
       END
```

```
      SUBROUTINE PLTROU(SAVE,K,ILOC,NLOC,IRUN)
      DIMENSION  SAVE(300,1),ILOC(1), IY(16), IR(2)
      IR(1)=IHEX(13,9,14,4,13,5,4,0)
      IR(2)=IHEX(13,5,13,6,4,11,4,0)
C   T
      IY(1) = IHEX(14,3,4,0,4,0,4,0)
C   U
      IY(2)=IHEX(14,4,4,0,4,0,4,0)
C   V
      IY(3) = IHEX(14,5,4,0,4,0,4,0)
C   W
      IY(4)=IHEX(14,6,4,0,4,0,4,0)
C   P
      IY(5)=IHEX(13,7,4,0,4,0,4,0)
C   Q
      IY(6)=IHEX(13,8,4,0,4,0,4,0)
C   R
      IY(7)=IHEX(13,9,4,0,4,0,4,0)
C   THTA
      IY(8)=IHEX(14,3,12,8,14,3,12,1)
C   PSI
      IY(9)=IHEX(13,7,14,2,12,9,4,0)
C   PHI
      IY(10)=IHEX(13,7,12,8,12,9,4,0)
C   X
      IY(11)=IHEX(14,7,4,0,4,0,4,0)
C   Y
      IY(12)=IHEX(14,8,4,0,4,0,4,0)
C   Z
      IY(13)=IHEX(14,9,4,0,4,0,4,0)
C   DR
      IY(14)=IHEX(12,4,13,9,4,0,4,0)
C   DS
      IY(15)=IHEX(12,4,14,2,4,0,4,0)
C   DB
      IY(16)=IHEX(12,4,12,2,4,0,4,0)
      DIV = 20.
      CALL SCALE(SAVE(1,1),6.0,K,1,DIV,1)
      ICTL=0
      CALL PLOT(0.0,.75,23)
      DO 80 I=1,NLOC
      J=ILOC(I)
      IF(J) 30,90,30
   30 CONTINUE
      CALL SCALE(SAVE(1,J),4.0,K,1,DIV,2)
      CALL AXIS(0.0,0.0,IY(1),-4,6.0,0.0,DIV,1)
      CALL AXIS(0.0,0.0,IY(J ),4,4.0,90.0,DIV,2)
      CALL SYMBOL(4.0,3.5,0.14,IR,0.0,8)
      AIRUN = IRUN
      CALL NUMBER(-0.0,-0.0,-0.0,AIRUN,0.0,-1)
      CALL LINE(SAVE(1,1),SAVE(1,J),K,1,0,0)
      IF(ICTL)60,50,60
   50 CALL PLOT(0.0,4.50,-23)
      ICTL=1
```

```
    GO TO 80
 60 CALL PLOT(8.5,-4.50,-23)
    ICTL=0
 80 CONTINUE
 90 IF(ICTL)110,100,110
100 CALL PLOT(0.0,-.75,23)
    RETURN
110 CALL PLOT(8.5,-5.25,23)
    RETURN
    END
```

```
      SUBROUTINE CONTR(THETA)
C
C TO CONTROL DS AND DR FOR DYNAMIC CONDITIONS
C
      DIMENSION TL(12), ILOC(16), Y(13)
      REAL IX,IY,IZ,IXY,IXZ,IYZ
      REAL K
C
      COMMON H, HMAX, HMIN, DH, FCT, TL, NGS, N, IS1, NPNT
C
      COMMON TLIM, RHOL2, RHOL3, RHOL4, RHOL5, WMB, ETA, ETAM1, ISW2
C
      COMMON       XQQ, XRR, XRP, XUD, XVR, XWQ, XUU, XVV,
     1             XWW, XDRDR, XDSDS, XDBDB, XVVE, XWWE, XDRDRE, XDSDSE
C
      COMMON       YRD, YRP, YPAP, YPQ, YQR, YVD, YVQ, YWP,
     1             YWR, YR, YP, YARDR, YVAR, YSTR,YV, YVAV,
     2             YVW, YDR, YRE, YVE, YVAVE, YDRE
C
      COMMON       ZQD, ZPP, ZRR, ZRP, ZWD, ZVR, ZVP, ZQ,
     1             ZAQDS, ZWAQ, ZSTR, ZW, ZWAW, ZAW, ZWW, ZVV,
     2             ZDS, ZDB, ZQE, ZWE, ZWAWE, ZDSE
C
      COMMON       AKPD, AKRD, AKQR, AKPQ, AKPAP, AKP, AKR, AKVD,
     1             AKVQ, AKWP, AKWR, AKSTR, AKV, AKVAV, AKVW, AKDR,
     2             AKSTRE
C
      COMMON       AMQD, AMPP, AMRR, AMRP, AMQAQ, AMWD,AMVR, AMVP,
     1             AMQ, AMAQDS, AMAWQ, AMSTR, AMW, AMWAW, AMAW, AMWW,
     2             AMVV, AMDS, AMDB, AMQE, AMWE, AMWAWE, AMDSE
C
      COMMON       ANRD, ANPD, ANPQ, ANQR, ANRAR, ANVD, ANWR, ANWP,
     1             ANVQ, ANP, ANR, ANARDR, ANAVR, ANSTR, ANV, ANVAV,
     2             ANVW, ANDR, ANRE, ANVE, ANVAVE, ANDRE
C
      COMMON       IX, IY, IZ, IXY, IXZ, IYZ
C
      COMMON       CW, CB, UC, XB, YB, ZB
C
      COMMON       DR, DS, DB, RHO, AL, AM
C
      COMMON       DRMAX, ETAHI, ETALO, A11, A12, A13
C
      COMMON       A21, A22, A23, A31, A32, A33
C
      COMMON       XG, YG, ZG
C
      COMMON       ILOC, IPLOT, IRUN, IOPEN, NPLT, IOPT
C
      COMMON       Y, TIME, R1, DELTMA, SWMAX, R2, DELTMI
C
      COMMON       DSE, DRE, ICYC, NS
      IF(NS)15,15,16
   15 RETURN
```

```
   16 CONTINUE
      GO TO(1001,1001,1003,1004,1005,1006,1007),NS
C
C CONTROL DS
C
 1001 IF(ISW2)21,20,21
   20 N1 = 2
    1 NN2 = 1
      NC2 = ((TIME*ICYC)/H) + .5
      NC3 = (ABS(DS     -DELTMA))*ICYC/ABS(R1*H) +.5
      NC5 = (ABS(DELTMI-DELTMA))*ICYC/ABS(R2*H) +.5
      GO TO 11
   21 GO TO (1,2,3,4,5,11),N1
C
C CYCLES TO START
C
    2 NN2 = NN2 + 1
      IF (NN2 - NC2) 11,11,7
C
C DS DOWN
C
    7 N1 = 3
      NN3 = 0
    3 NN3 = NN3 +1
      DS = DS   + H*R1/ICYC
      IF(NN3 - NC3) 11,08,8
C
C DS LEVEL
C
    8 N1 = 4
      GO TO 11
    4 IF (ABS(THETA) - SWMAX) 11,9,9
C
C DS UP
C
    9 N1 = 5
      NN5 = 0
    5 NN5 = NN5 + 1
      DS = DS   + H*R2/ICYC
      IF (NN5 -NC5) 11,10,10
C
C DS LEVEL
C
   10 N1 = 6
C
   11 IF (NS - 2)13,1003,1003
   13 CONTINUE
      GO TO 2000
C
C CONTROL DR + AUTOPILOT
C
 1003 IF (ISW2) 301,300,301
  300 ZC = Y(12)
      SDOT1 = 0.
```

```
      DDR = ABS(DRMAX)
      NC10= ((TIME * ICYC)/H) + .5
      NC6 = .85 * DDR * ICYC/(.08726*H) +.5
      NC7 = .08 * DDR * ICYC/(.01336*H) +.5
      NC8 = .04 * DDR * ICYC/(.006*H) +.5
      NC9 = .03 * DDR * ICYC/( .001064*H)+.5
      NN10= 1
      N2 = 2
      IF(DRMAX)313,314,314
  313 R6 = -.08726
      R7 = -.01336
      R8 = -.006
      R9 = -.001064
      GO TO 350
  314 R6 = .08726
      R7 = .01336
      R8 = .006
      R9 = .001064
      GO TO 350
  301 GO TO (300,302,303,304,305,306,350),N2
  302 NN10 = NN10 + 1
      IF (NN10-NC10)350,350,308
C
C FROM 0 TO .85 OF DRMAX
C
  308 N2 = 3
      NN6 = 0
  303 NN6 = NN6 + 1
      DR = DR  + H*R6/ICYC
      IF (NN6 - NC6) 350,309,309
C
C FROM .85 TO .93 OF DRMAX
C
  309 N2 = 4
      NN7 = 0
      GO TO 350
  304 NN7 = NN7 + 1
      DR = DR  + H*R7/ICYC
      IF (NN7 - NC7) 350,310,310
C
C FROM .93 TO .97 OF DRMAX
C
  310 N2 = 5
      NN8 = 0
      GO TO 350
  305 NN8 = NN8 + 1
      DR = DR  + H*R8/ICYC
      IF (NN8 - NC8) 350,311,311
C
C FROM .97 TO 1. OF DRMAX
C
  311 N2 = 6
      NN9 = 0
      GO TO 350
```

```
  306 NN9 = NN9 + 1
      DR = DR  + H*R9/ICYC
      IF (NN9 - NC9) 350,312,312
C
C LEVEL, DRMAX
C
  312 N2 = 7
  350 IF(NS-2)2000,2000,352
C
C AUTOPILOT
C
  352 DSC=.008*(ZC-Y(12))+3.5*Y(7)+.012*(Y(1)*SIN(Y(7))-Y(3)*COS(Y(7)))
     1+2.*Y(5)
  103 IF (DSC) 110,107,107
  107 IF (DSC - .436) 101,108,108
  110 IF (DSC + .436) 109,101,101
  108 DSC = .436
      GO TO 101
  109 DSC = -.436
  101 SDOT = 3 * (DSC -DS)
      DS = DS + .5 * H/ICYC * (3. * SDOT - SDOT1)
      SDOT1 = SDOT
      DB = -DS
  351 CONTINUE
      GO TO 2000
C
C CONTROL DS (IMPULSE), LONGITUDINAL
C
 1004 IF (ISW2)401,400,401
  400 IF(ICYC-1)411,411,412
  411 N4=0
      NTST=1
      NMOD=8
      GO TO 401
  412 N4=-1
      NTST=3
      NMOD=32
  401 IF(N4-NTST)403,402,403
  402 DS = DSF
  403 IF(MOD(N4,NMOD))410,406,410
C
C PUNCH THETA AND TIME FOR FREQUENCY STUDY (LONGITUDINAL)
C
  406 WRITE(2,408)Y(7),Y(13)
  408 FORMAT(2E15.7)
  410 N4 = N4 + 1
      GO TO 2000
C
C CONTROL DR (IMPULSE), LATERAL
C
 1005 IF (ISW2)501,500,501
  500 IF(ICYC-1)511,511,512
  511 N5 = 0
      NTST =1
```

```
      NMOD=8
      GO TO 501
  512 N5=-1
      NTST=3
      NMOD=32
  501 IF(N5-NTST)503,502,503
  502 DR = DRF
  503 IF(MOD(N5,NMOD))510,506,510
C
C PUNCH PHI AND TIME FOR FREQUENCY STUDY(LATERAL)
C
  506 WRITE(2,408) Y(9), Y(13)
  510 N5 = N5 + 1
      GO TO 350
C
C CONTROL ACCEL/DECEL + AUTOPILOT
C
 1006 IF(ISW2)601,600,601
  600 N6=1
      ISW6=0
      NN11=1
      TLIM=     10.*TIME+60.
      ZC = Y(12)
      SDOT1 = 0.
      NC11=60*(ICYC/H)
      NC12=TIME*ICYC/H
      UC=0.
      GO TO 352
  601 GO TO(602,603,604,605,606,607,352),N6
C
      UC=0.
  602 NN11=NN11+1
      IF(NN11-NC11)352,352,608
C
  608 N6=2
      UC=8.445
      NN12=0
  603 NN12=NN12+1
      IF(NN12-NC12)352,352,609
C
  609 IF(ISW6)617,616,617
  617 N6=7
      UC=0.
      GO TO 352
C
  616 N6=3
      UC=16.89
      NN12=0
  604 NN12=NN12+1
      IF(NN12-NC12)352,352,610
C
  610 IF(ISW6)618,615,618
  618 GO TO 608
C
```

```
  615 N6=4
      UC=25.335
      NN12=0
  605 NN12=NN12+1
      IF(NN12-NC12)352,352,611
C
  611 IF(ISW6)619,614,619
  619 GO TO 616
C
  614 N6=5
      UC=33.78
      NN12=0
  606 NN12=NN12+1
      IF(NN12-NC12)352,352,612
C
  612 IF(ISW6)620,621,620
  620 GO TO 615
C
  621 N6=6
      UC=42.225
      NN12=0
  607 NN12=NN12+1
      IF(NN12-NC12)352,352,613
C
  613 ISW6 = 1
      GO TO 614
C
C       CONTROL MAXIMUM ACCEL/DECEL + AUTOPILOT
C
 1007 IF(ISW2)701,700,701
  700 N7=1
      NN13=1
      TLIM=60.+2.*TIME
      NC13=60*ICYC/H
      NC14=TIME*ICYC/H
      SDOT1=0.
      ZC=Y(12)
      UC=0.
      GO TO 352
  701 GO TO(702,703,352),N7
C
C     UC=0.
  702 NN13=NN13+1
      IF(NN13-NC13)352,352,705
C
  705 N7=2
      UC=42.225
      NN14=0
  703 NN14=NN14+1
      IF(NN14-NC14)352,352,706
C
  706 N7=3
      UC=0.
      GO TO 352
```

```
 2000 RETURN
      END
/*
/&
```

03/10/69 AUTOFLOW CHART SET - FB029 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES

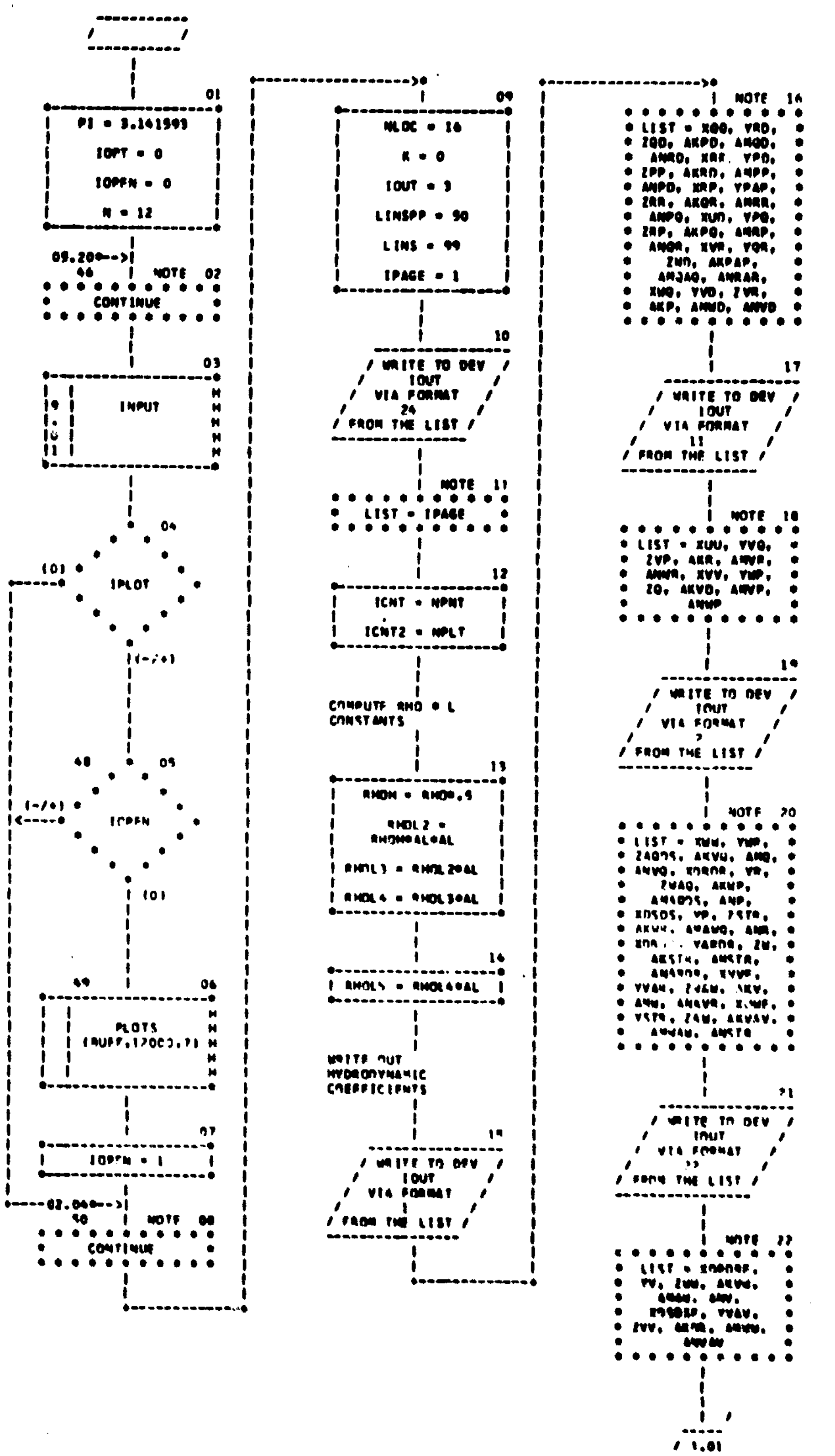
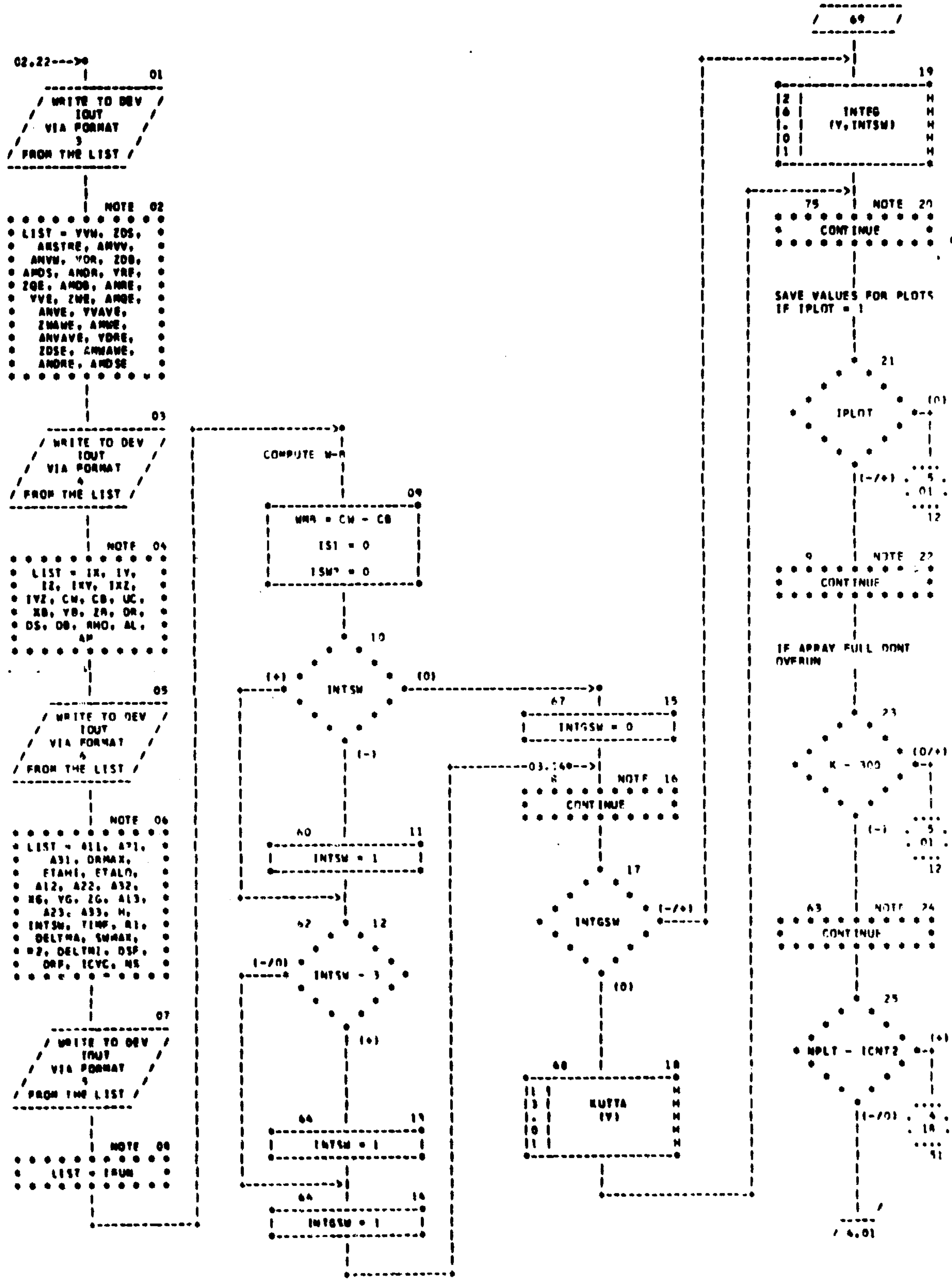
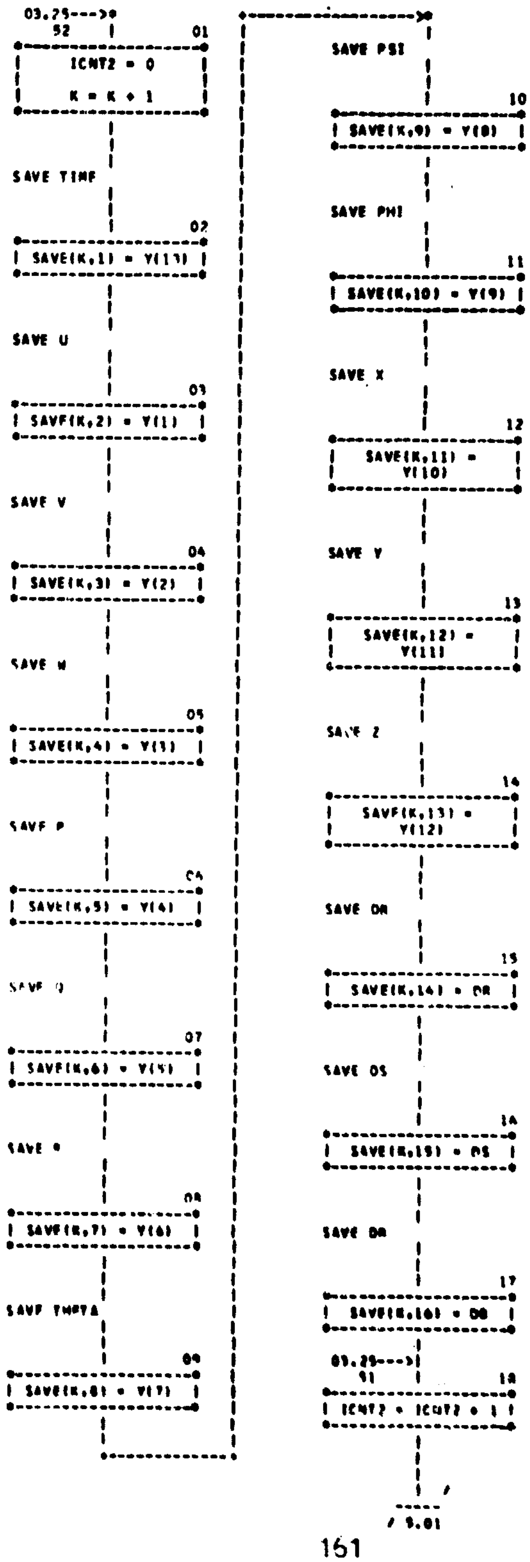


CHART TITLE - PROCEDURES



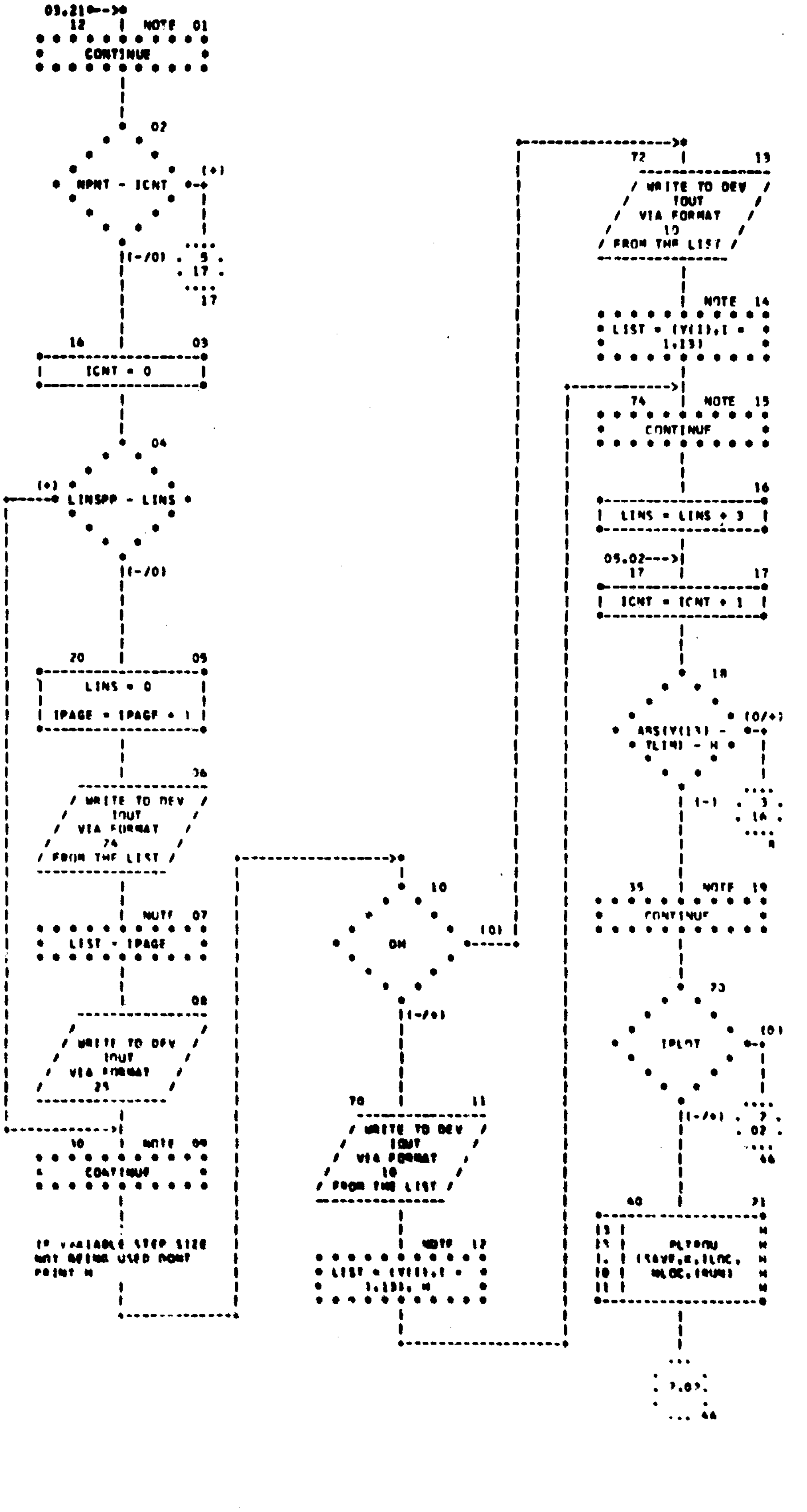
03/10/69 AUTOFLOW CHART SET - E0920 NAVTRADEVCEN 69-C-0050-2

CHART TITLE - PROCEDURES



03/10/69 AUTOFLOW CHART SET - E8920 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES



03/10/69 AUTOFLOW CHART SET - E8920 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE INPUT

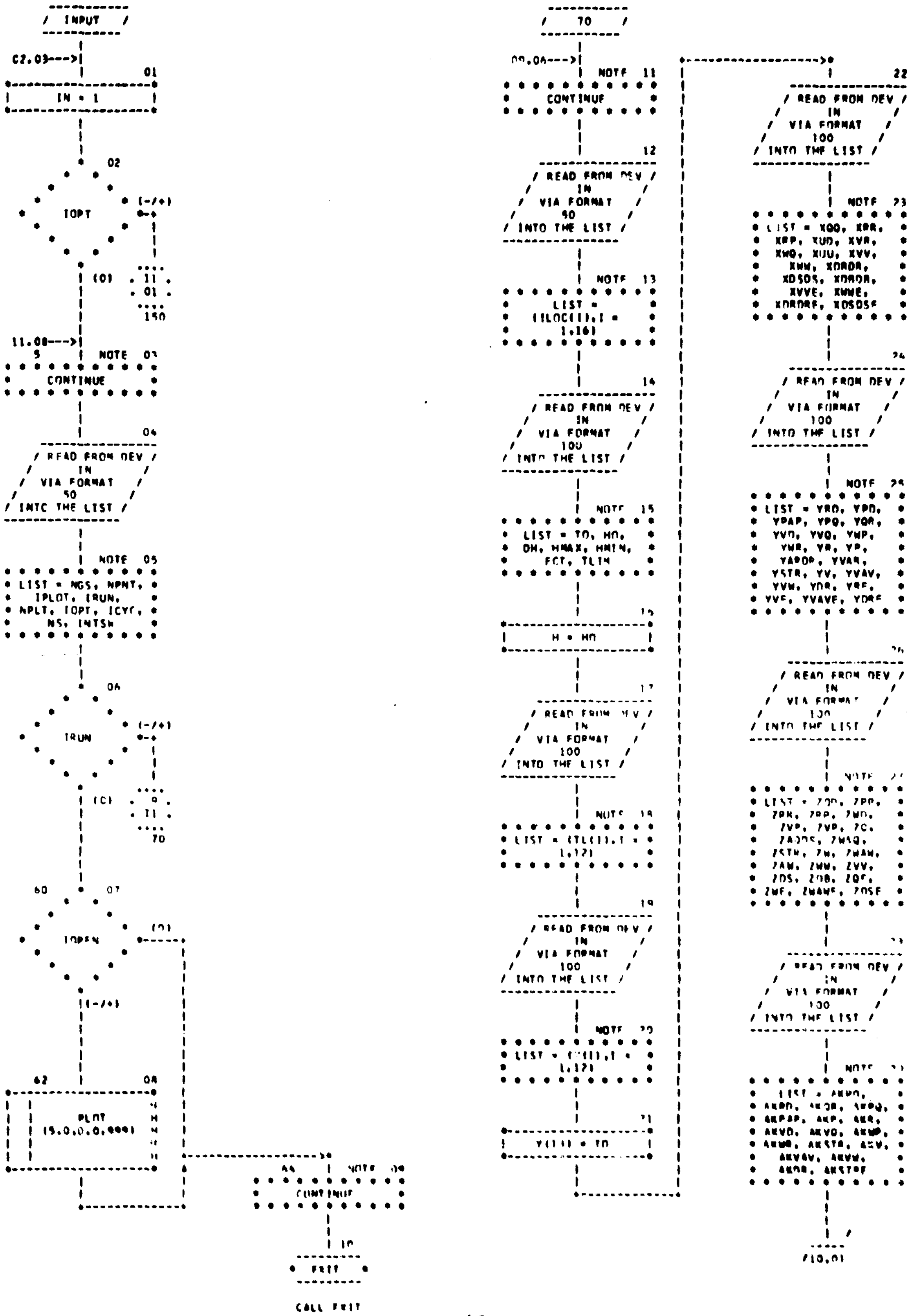


CHART TITLE - SUBROUTINE INPUT

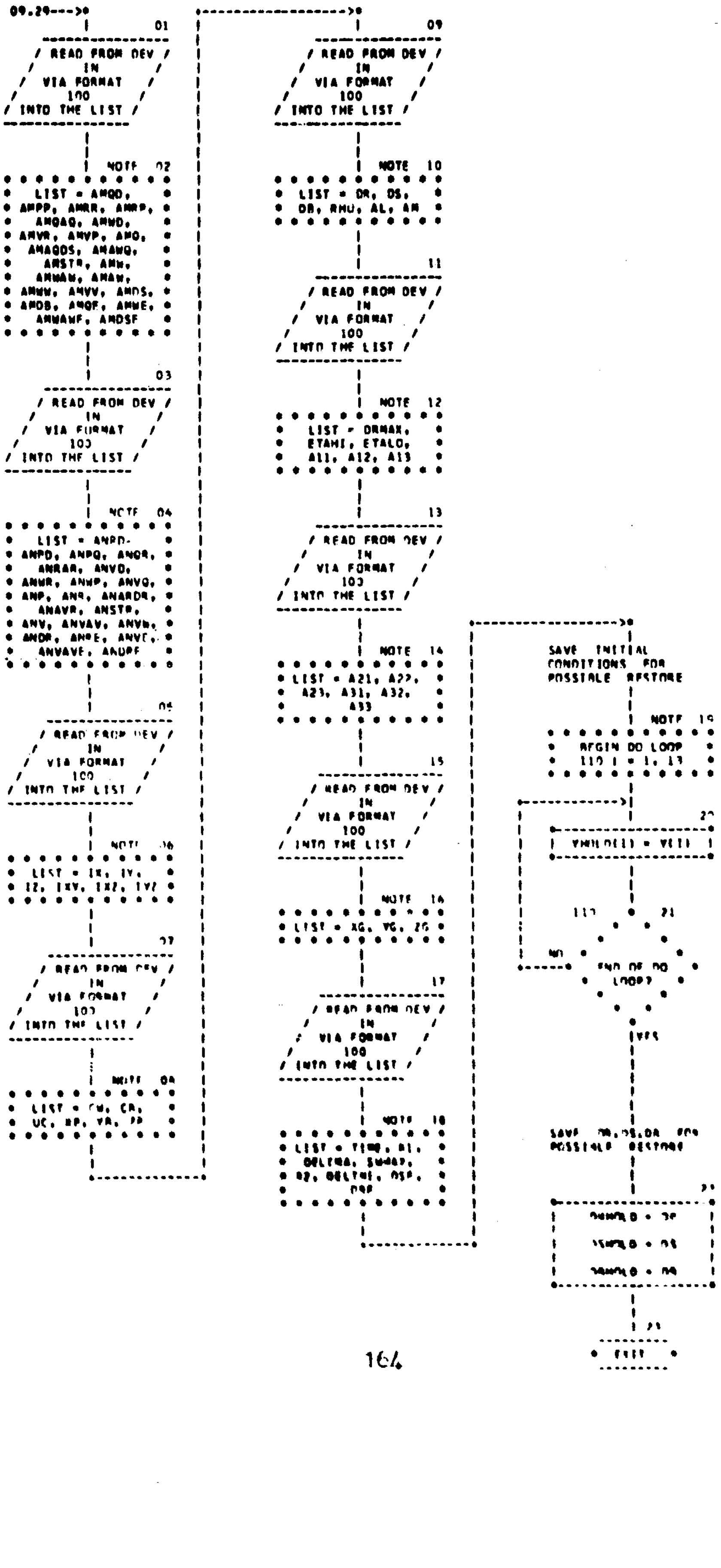
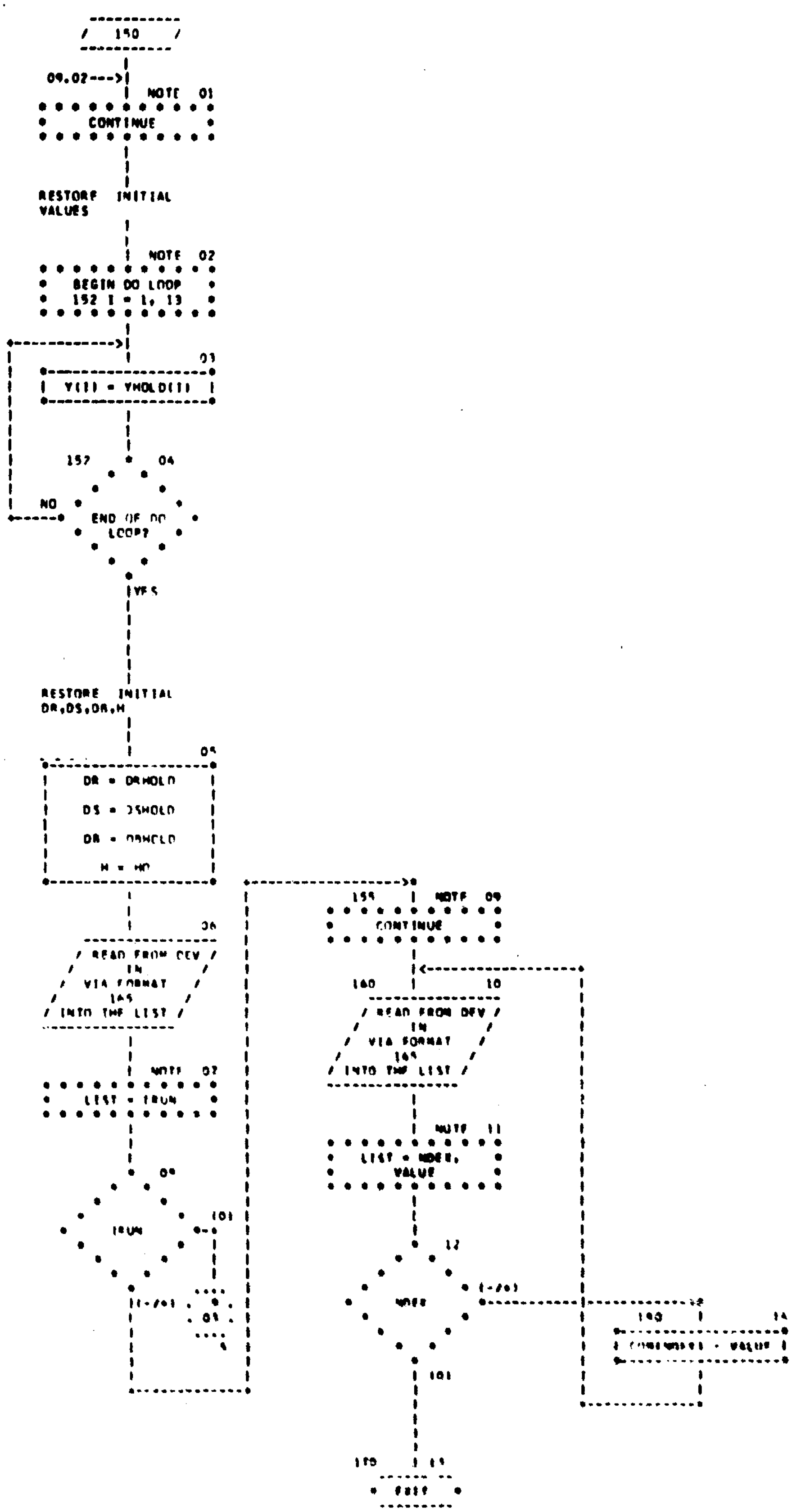
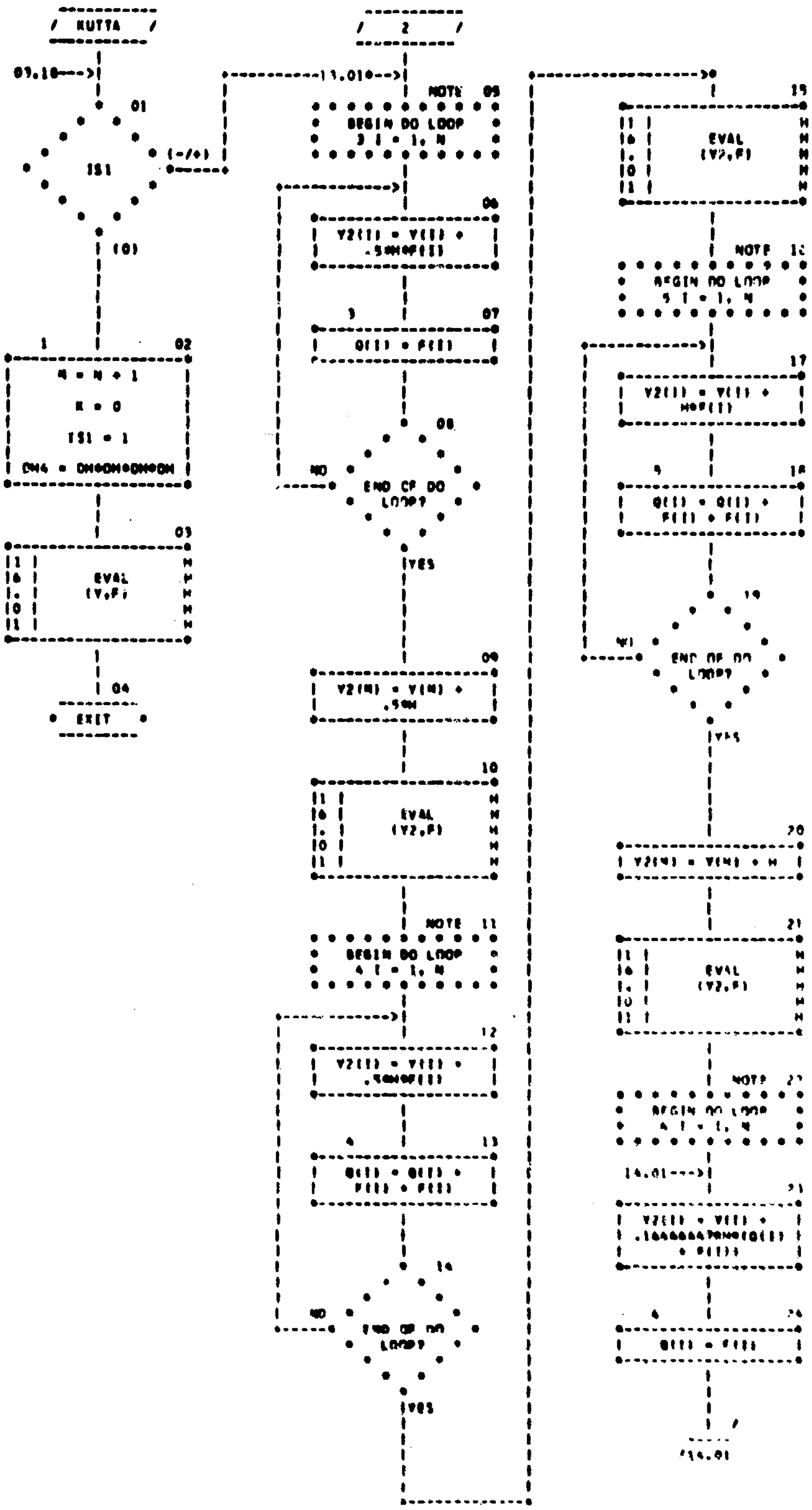


CHART TITLE - SUBROUTINE INPUT



03/10/69 AUTOFLOW CHART SET - EB920 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE KUTTA(Y)



03/10/69 AUTOFLOW CHART SET - EB920 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE KUTTA(Y)

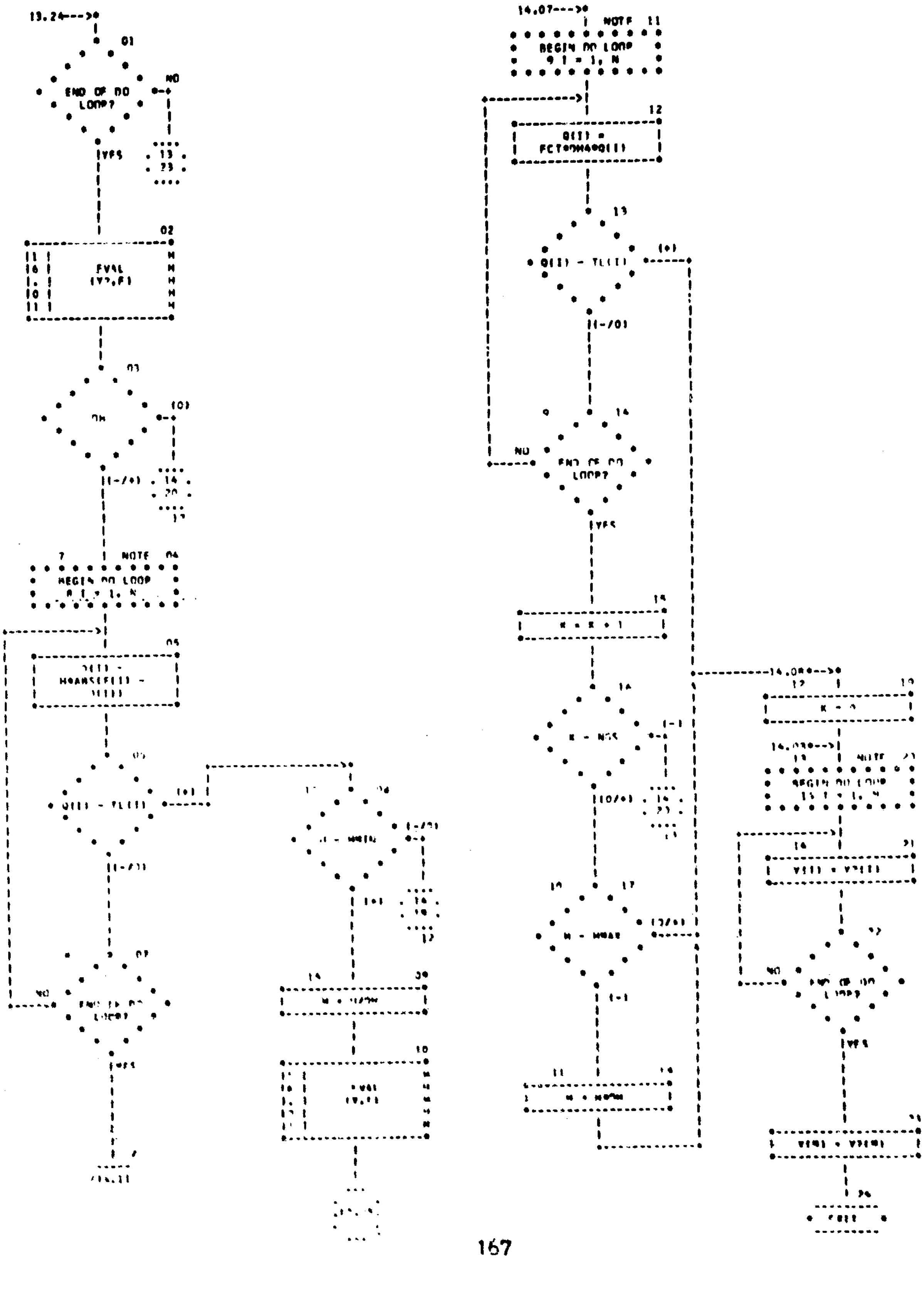
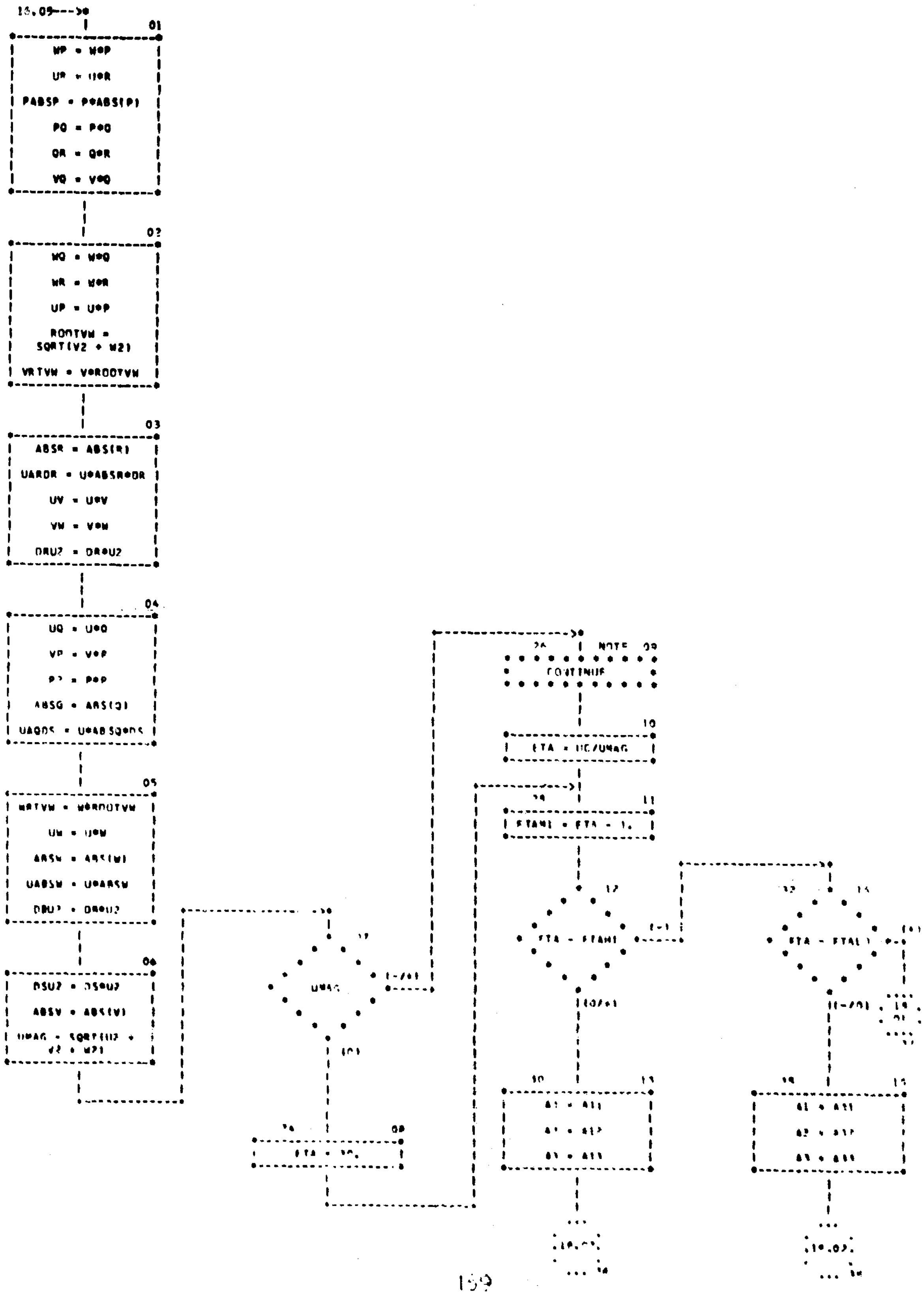


CHART TITLE - SUBROUTINE EVAL(YI,F)

```
                 -----------
                /   EVAL    /
                -----------

               13,03*-->*

               PULL PRESENT VALUES
               OF VARIABLES OUT OF
               ARRAY YI

                        *
                        I          01
               *--------------------*
               I     U = YI(1)      I
               I                    I
               I     V = YI(2)      I
               I                    I
               I     W = YI(3)      I
               I                    I
               I     P = YI(4)      I
               I                    I
               I     Q = YI(5)      I
               I                    I
               I     R = YI(6)      I
               *--------------------*
                        I
                        I
                        I          02
               *--------------------*
               I  THETA = YI(7)     I
               I                    I
               I   PSI = YI(8)      I
               I                    I
               I   PHI = YI(9)      I
               I                    I
               I    X = YI(10)      I
               I                    I
               I    Y = YI(11)      I
               I                    I
               I    Z = YI(12)      I
               *--------------------*
                        I
                        I
                        I          03
               *--------------------*
               I3 I                 H
               I6 I     CONTR       H
               I. I    (THETA)      H
               I0 I                 H
               I1 I                 H
               *--------------------*
                        I
                        I
               COMPUTE QUANTITIES TO
               BE USED MORE THAN
               ONCE
                        I
                        I          04
               *--------------------*
               I     VR = V*R       I
               I                    I
               I     Q2 = Q*Q       I
               I                    I
               I     R2 = R*R       I
               I                    I
               I     P2 = P*P       I
               I                    I
               I     U2 = U*U       I
               I                    I
               I     V2 = V*V       I
               *--------------------*
                        I
                        I          05
               *--------------------*
               I     W2 = W*W       I
               I                    I
               I   DR2 = DR*DR      I
               I                    I
               I DR2U2 = DR2*U2     I
               I                    I
               I   DS2 = DS*DS      I
               I                    I
               I DS2U2 = DS2*U2     I
               *--------------------*
                        I
                        I
                        I  /
                      -----
                     /19.01
```

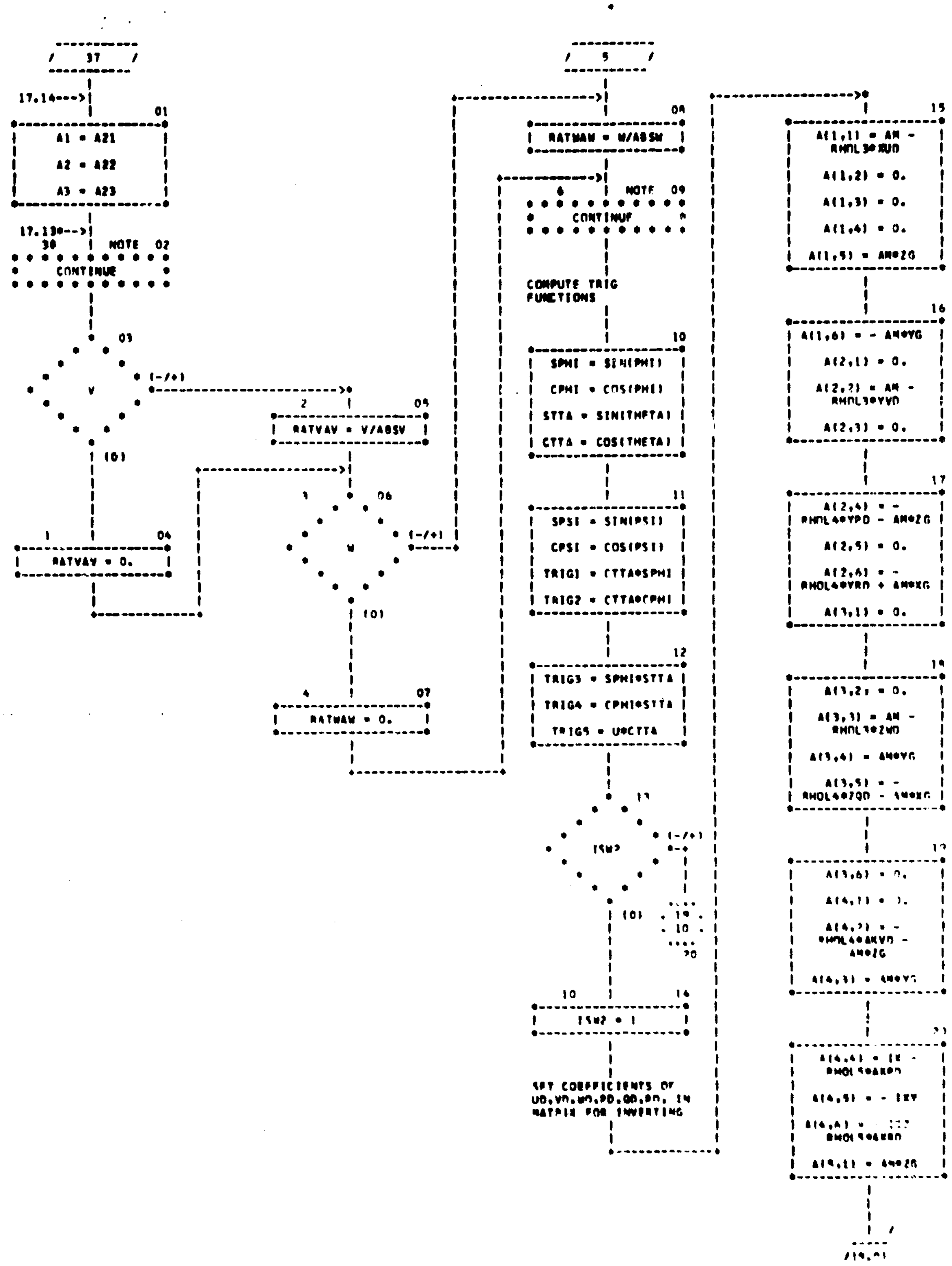
03/10/69 AUTOFLOW CHART SET - E8920 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE EVAL(Y1,F)



03/10/69 AUTOFLOW CHART SET - EB920 NAVTRADEVCEN 68-C-0050-2

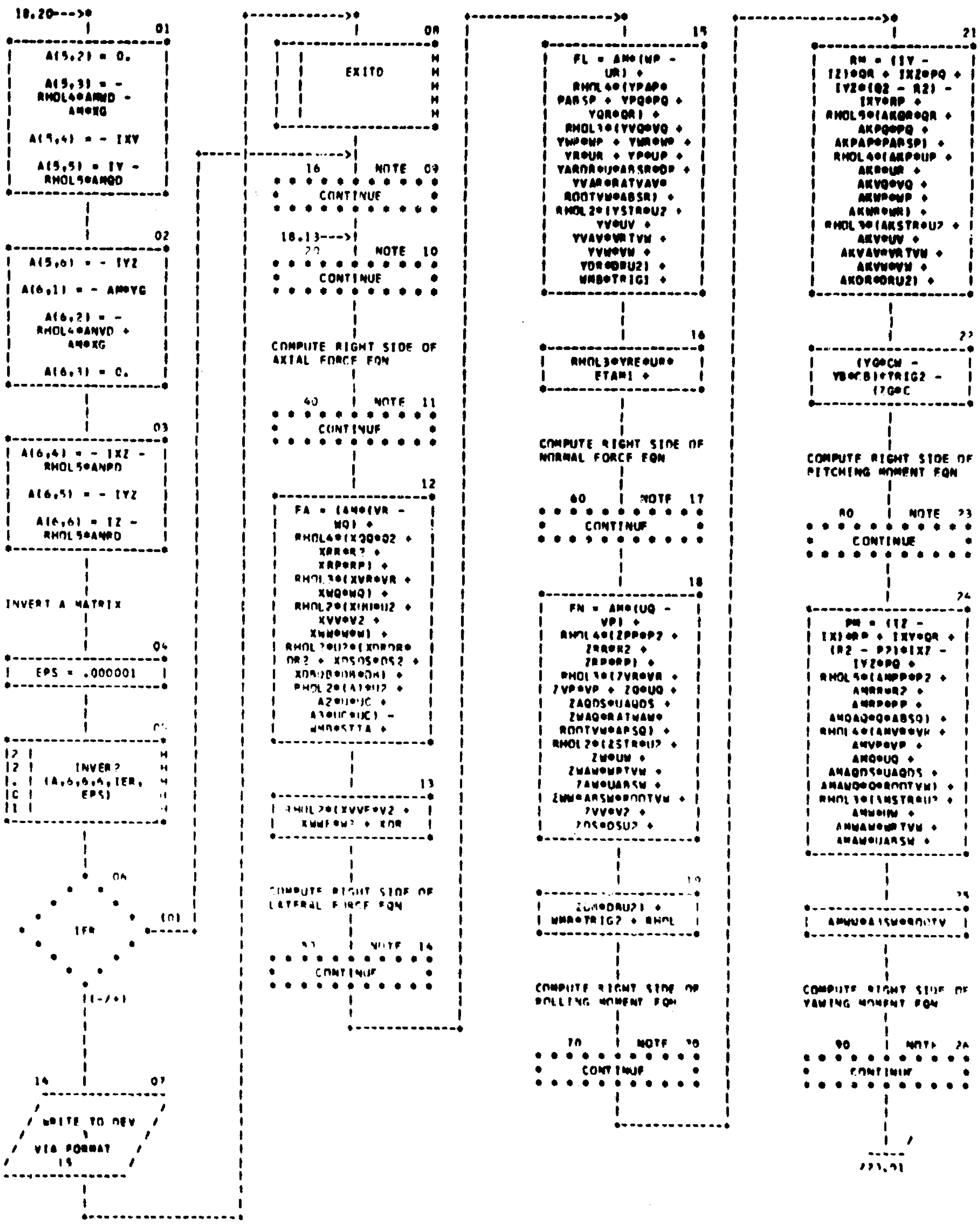
CHART TITLE - SUBROUTINE FVAL(V1,F)



03/10/69

AUTOFLOW CHART SET - FB020   NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE EVAL(YI,F)



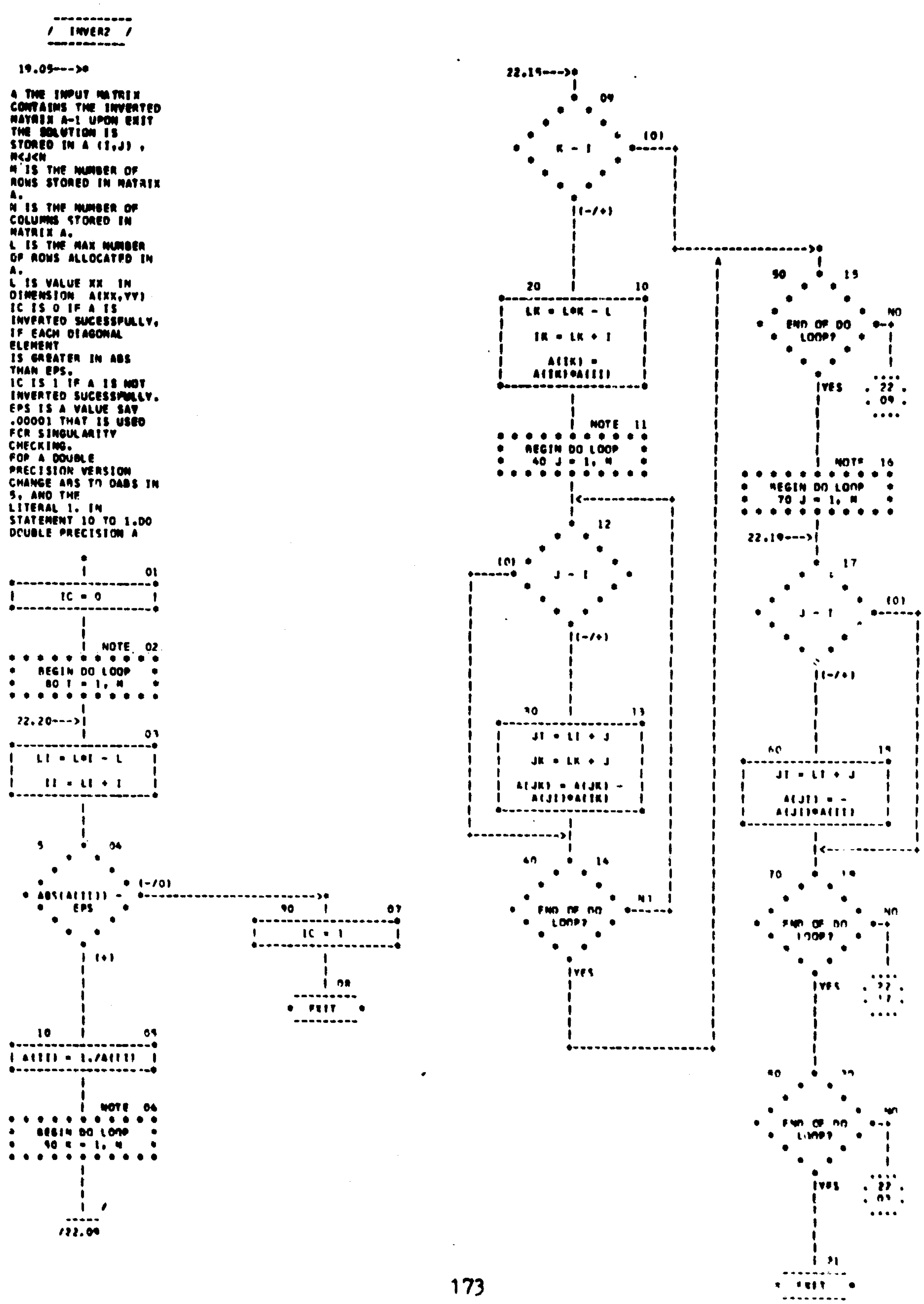
09/10/69 AUTOFLOW CHART SET - 88920 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE EVAL(Y1,F)

```
 19.76--->*
          |               01
*----------------------------*
|     VN = (IX -             |
|  IY)*PQ + IYZ*RP +         |
|   (P2 - Q2)*IXY -          |
|      IXZ*QR +              |
| RHOL5*(ANPQ*PQ +           |
|      ANQR*QR +             |
|  ANRAR*R*ABSR) +           |
| RHOL4*(ANWR*WR +           |
|      ANWP*WP +             |
|      ANVQ*VQ +             |
| ANP*UP + ANR*UR +          |
|   ANARDR*UARDR +           |
| ANAVR*R*ROOTVW) +          |
| RHOL3*(ANSTR*U2 +          |
|      ANV*UV +              |
|    ANVAV*VRTVW +           |
|      ANVW*VW +             |
*----------------------------*
          |
          |
          |               02
*----------------------------*
|      ANDR*DRU2             |
*----------------------------*
          |
          |
MULTIPLY TO GET UD,
VD, WD, PD, QD, RD
          |
          |               03
*----------------------------*
|2 |                        H
|4 |       MATMPY           H
|. | (A,B,F,6,6,1,          H
|0 |     6,6,6,1)           H
|1 |                        H
*----------------------------*
          |
          |
COMPUTE KINEMATICS -
THETA DOT , PSI DOT,
PHI DOT
          |
          |               04
*----------------------------*
|  F(7) = Q*CPHI -           |
|      R*SPHI                |
|                            |
| F(8) = (Q*SPHI +           |
|   R*CPHI)/CTTA             |
|                            |
|    F(9) = P +              |
|     F(8)*STTA              |
*----------------------------*
          |
          |
COMPUTE  X DOT , Y
DOT , Z DOT
          |
          |               05
*----------------------------*
|     F(10) =                |
|   TRIG8*CPSI +             |
|  V*(TRIG3*CPSI -           |
|   CPHI*SPSI) +             |
|  W*(TRIG4*CPSI +           |
|    SPHI*SPSI)              |
*----------------------------*
          |
          |
          *------------------------------>*
                                          |          06
                         *----------------------------*
                         |      F(11) =               |
                         |   TRIG9*SPSI +             |
                         |  V*(TRIG7*SPSI +           |
                         |   CPHI*CPSI) +             |
                         |  W*(TRIG6*SPSI -           |
                         |    SPHI*CPSI)              |
                         |                            |
                         |     F(12) = -              |
                         |     U*STTA +               |
                         | V*TRIG1 + W*TRIG2          |
                         *----------------------------*
                                          |
                                          |
                                          | 07
                                      ----------
                                    *   EXIT   *
                                      ----------
```

03/10/69 AUTOFLOW CHART SET - FB920 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE INVER2(A,M,N,L,IC,EPS)



03/10/69 AUTOFLOW CHART SET - ER920 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE MATMPY(A,B,C,M,N,L,MA,MB,MC,IOPT)

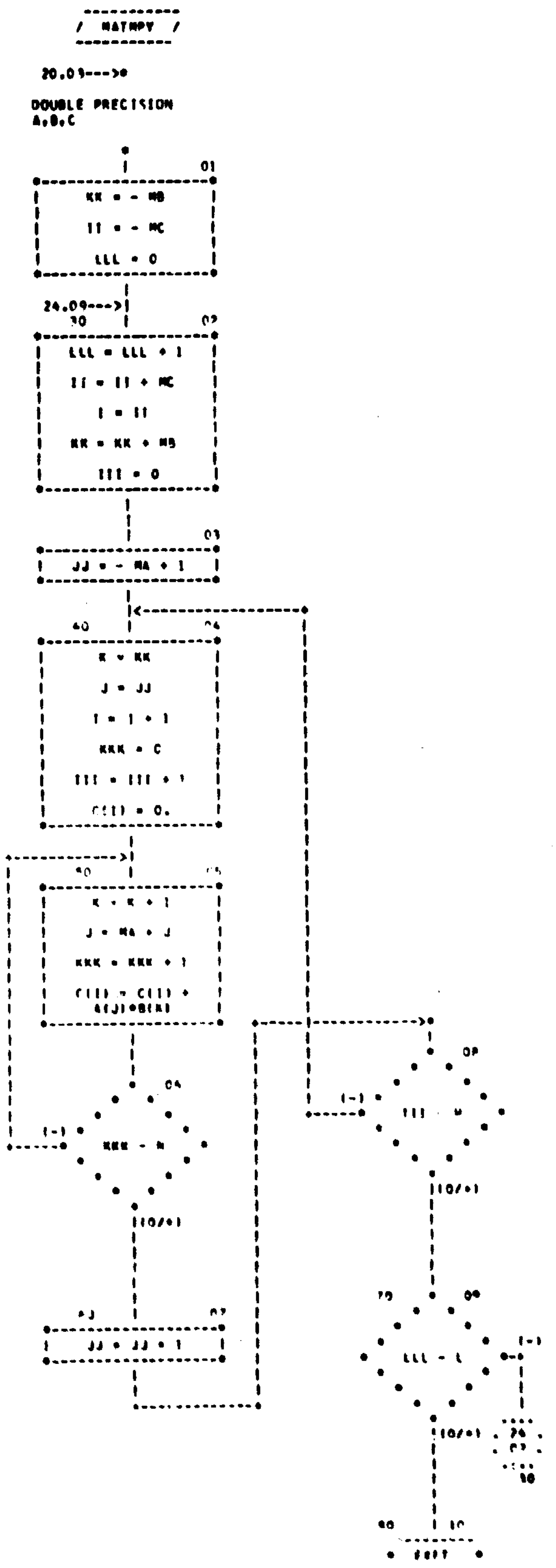


CHART TITLE - SUBROUTINE INTEG(Y,INTSW)

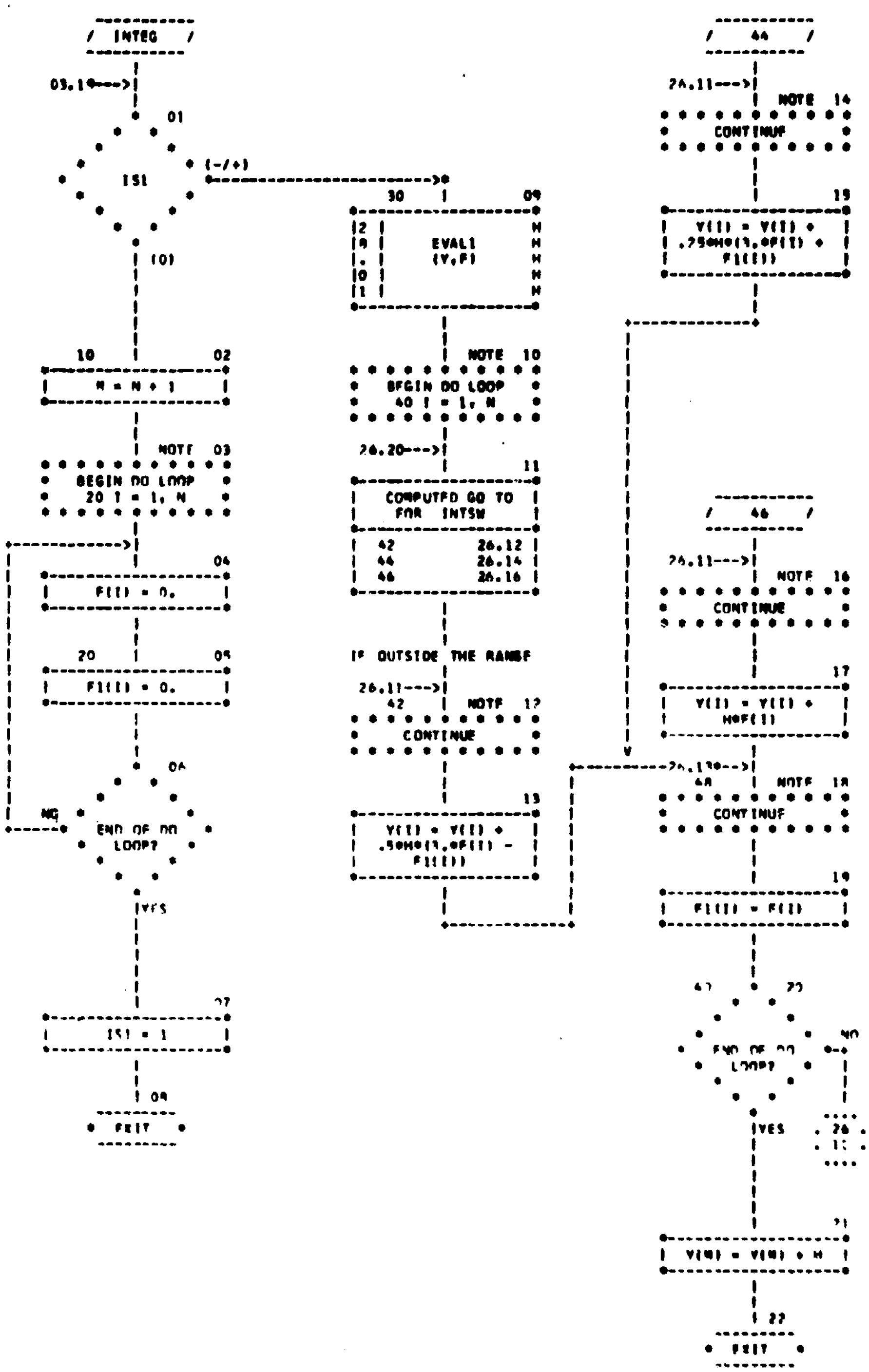


CHART TITLE - SUBROUTINE EVAL1(VI,F)

```
 ------------
/ EVAL1    /
 ------------

26.09--->*

PULL PRESENT VALUES
OF VARIABLES OUT OF
ARRAY VI

          *
          |              01
*------------------------*
|      U = VI(1)         |
|      V = VI(2)         |
|      W = VI(3)         |
|      P = VI(4)         |
|      Q = VI(5)         |
|      R = VI(6)         |
*------------------------*
          |
          |              02
*------------------------*
|   THETA = VI(7)        |
|   PSI  = VI(8)         |
|   PHI  = VI(9)         |
|     X = VI(10)         |
|     Y = VI(11)         |
|     Z = VI(12)         |
*------------------------*
          |
          |              03
*------------------------*
|3 |                     H
|6 |     CONTR           H
|. |    (THETA)          H
|0 |                     H
|1 |                     H
*------------------------*
          |
          |
COMPUTE QUANTITIES TO
BE USED MORE THAN
ONCE
          |
          |              04
*------------------------*
|      VR = V*R          |
|      Q2 = Q*Q          |
|      R2 = R*R          |
|      PR = R*P          |
|      U2 = U*U          |
|      V2 = V*V          |
*------------------------*
          |
          |              05
*------------------------*
|      W2 = W*W          |
|     DR2 = DR*DR        |
|   DR2U2 = DR2*U2       |
|     DS2 = DS*DS        |
|   DS2U2 = DS2*U2       |
*------------------------*
          |
          |  /
        -----
       /29.01
```

CHART TITLE - SUBROUTINE EVAL(IV,F)

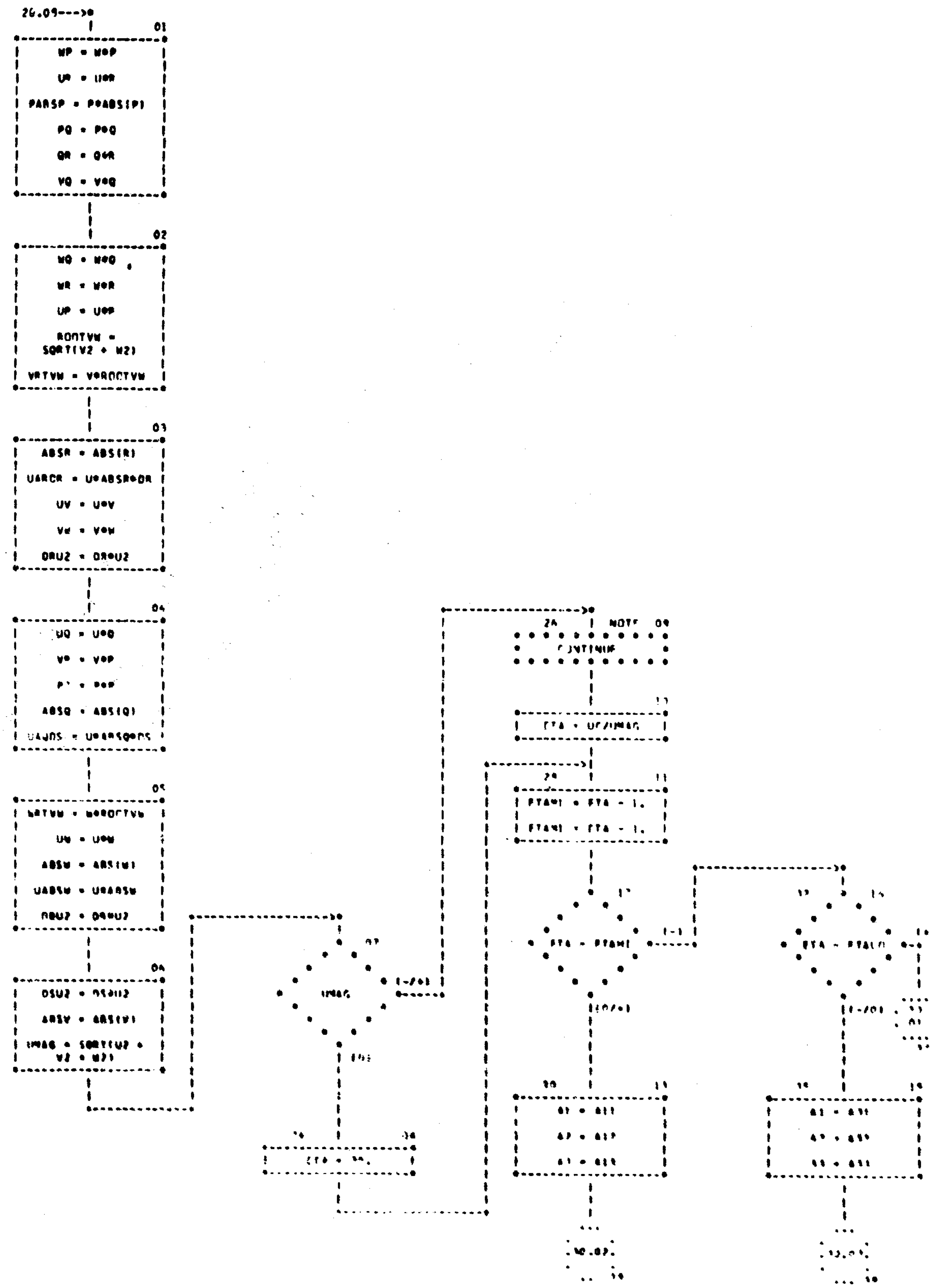


CHART TITLE - SUBROUTINE VSLI(VI,F)

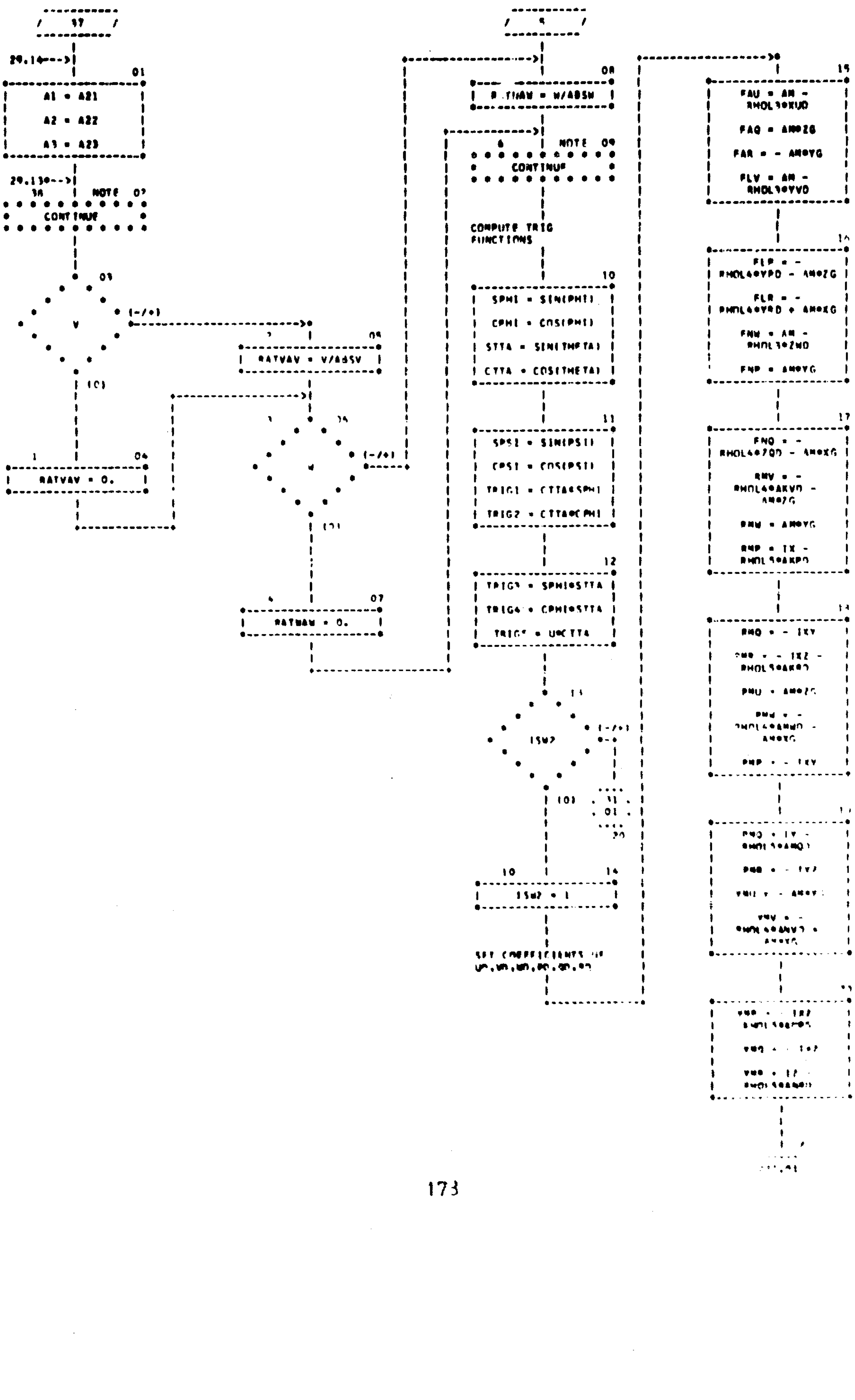


CHART TITLE - SUBROUTINE EVAL(IY1,F)

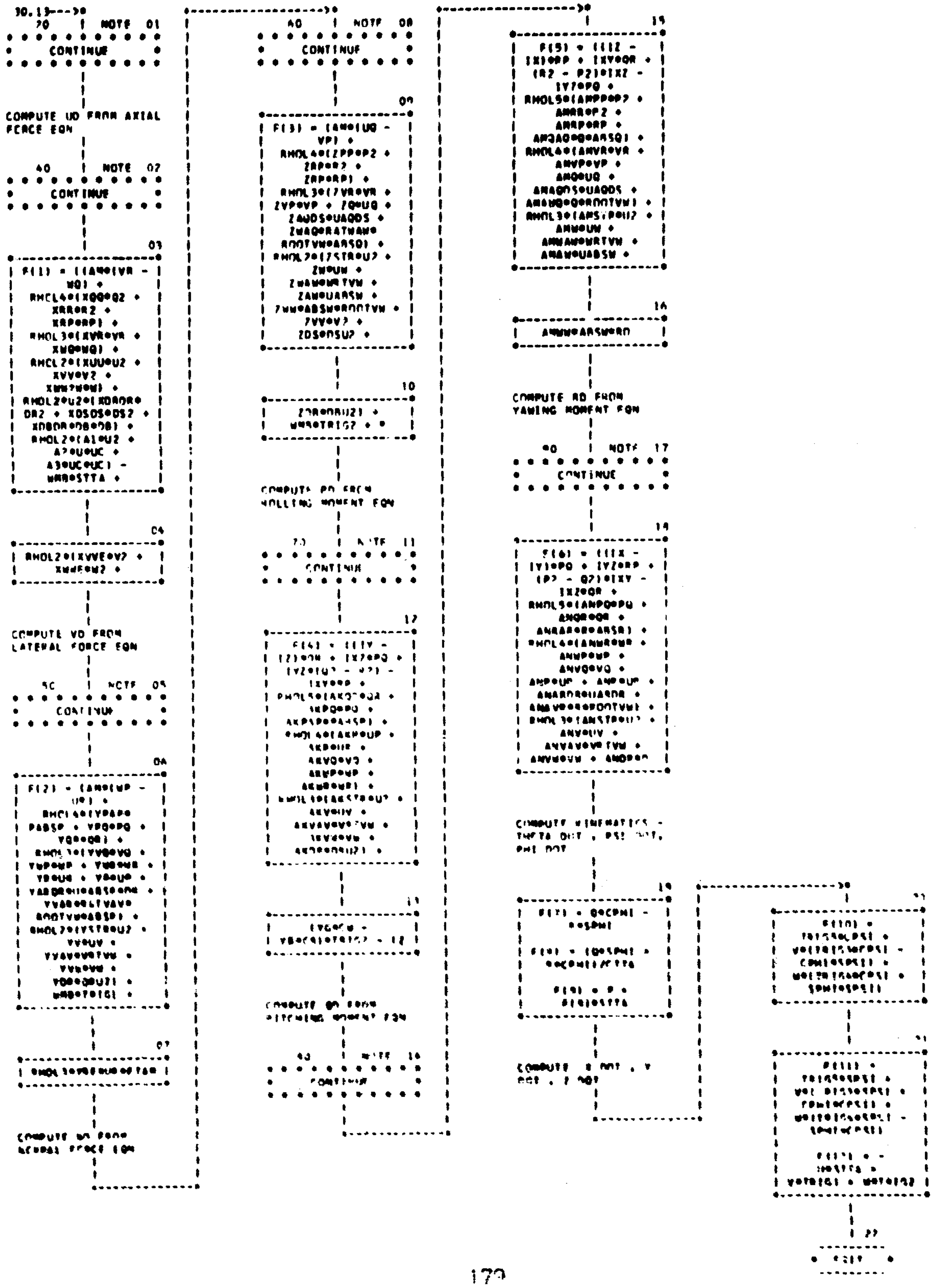
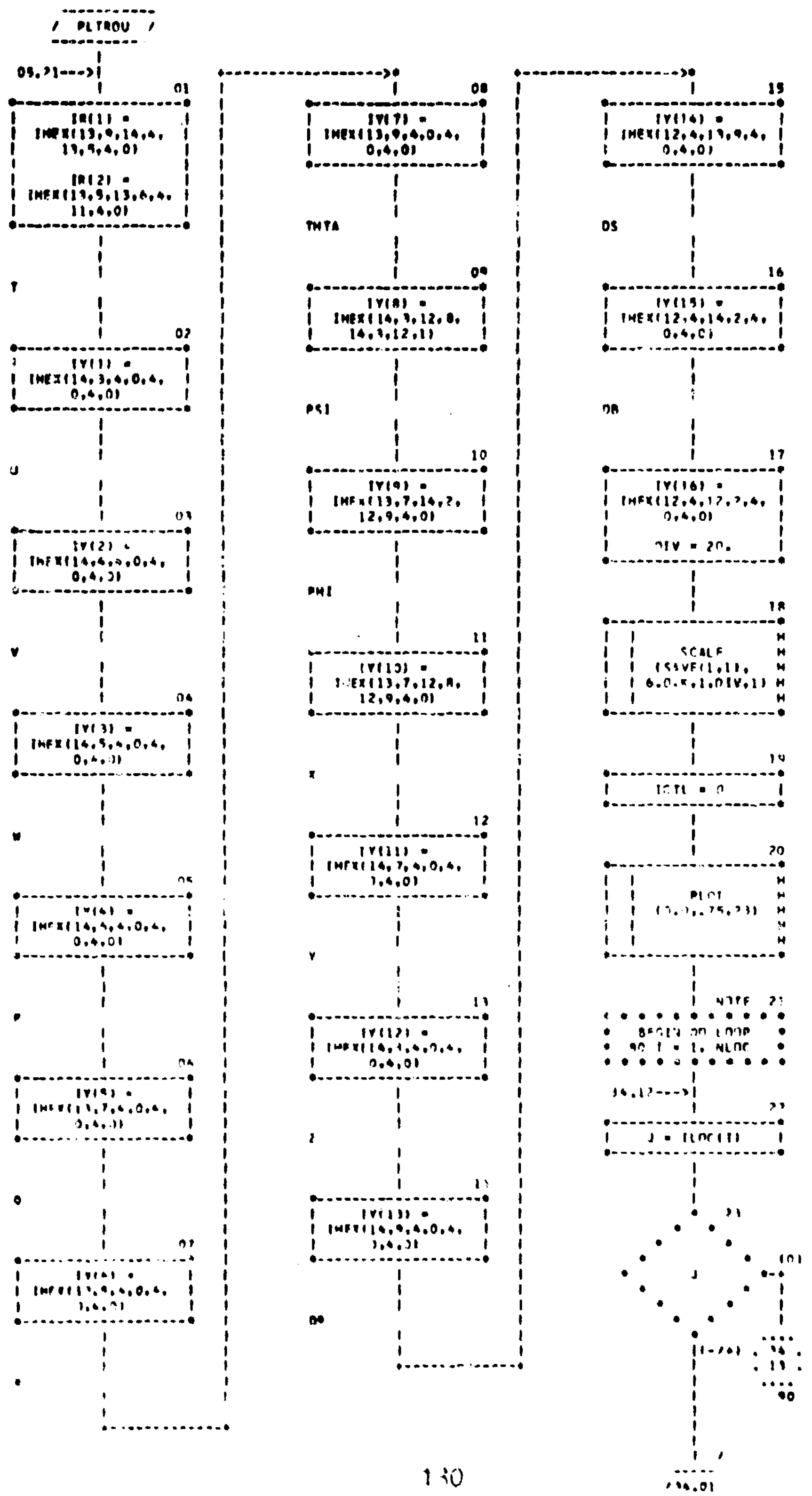


CHART TITLE - SUBROUTINE PLTROU(SAVE,K,ILOC,NLOC,IRUN)



03/10/69 AUTOFLOW CHART SET - EB920 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE PLTROU(SAVE,K,ILOC,NLOC,IRUN)

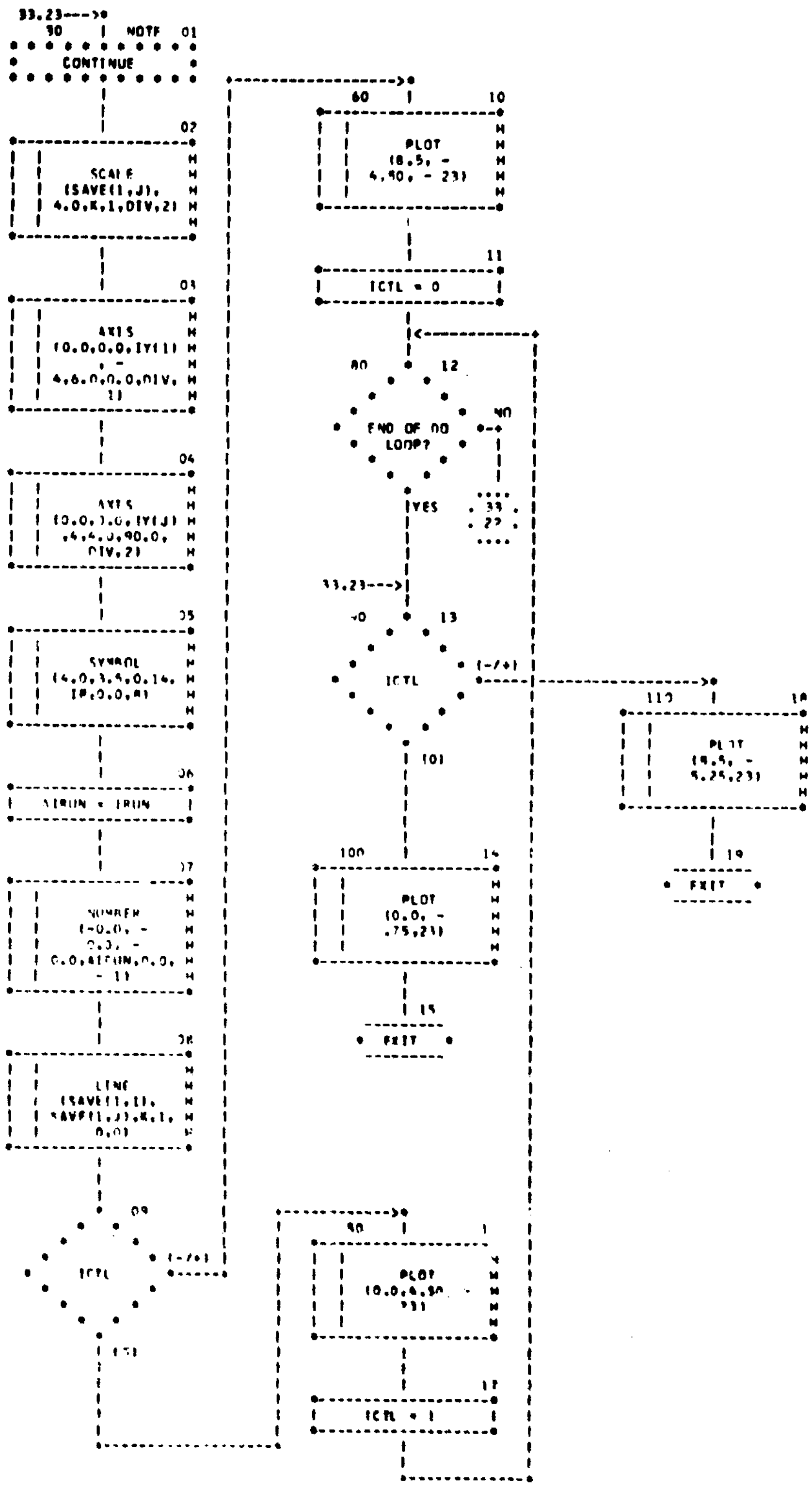
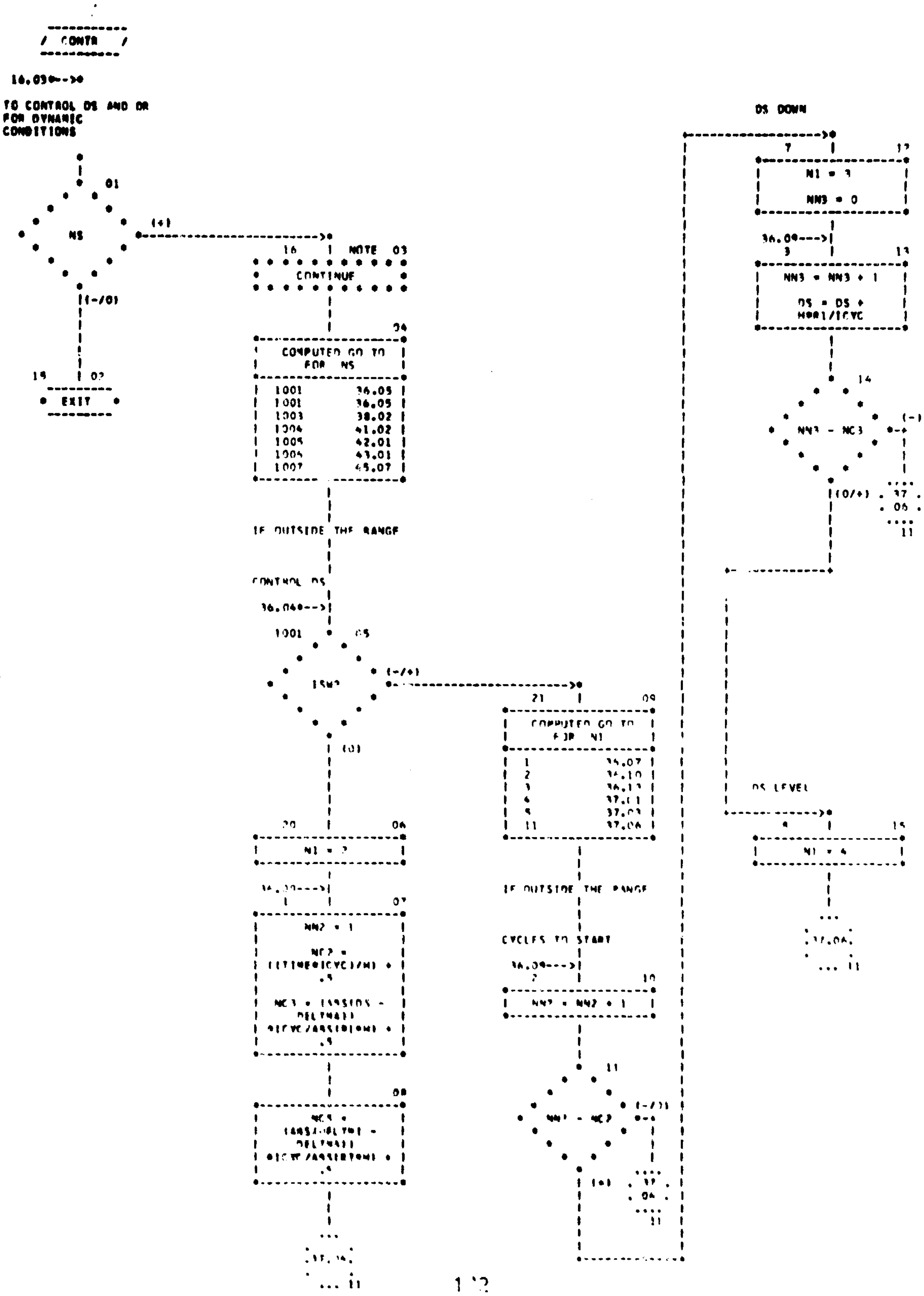


CHART TITLE - SUBROUTINE CONTR(THETA)



03/10/69 AUTOFLOW CHART SET - FR020 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE CONTR(THETA)

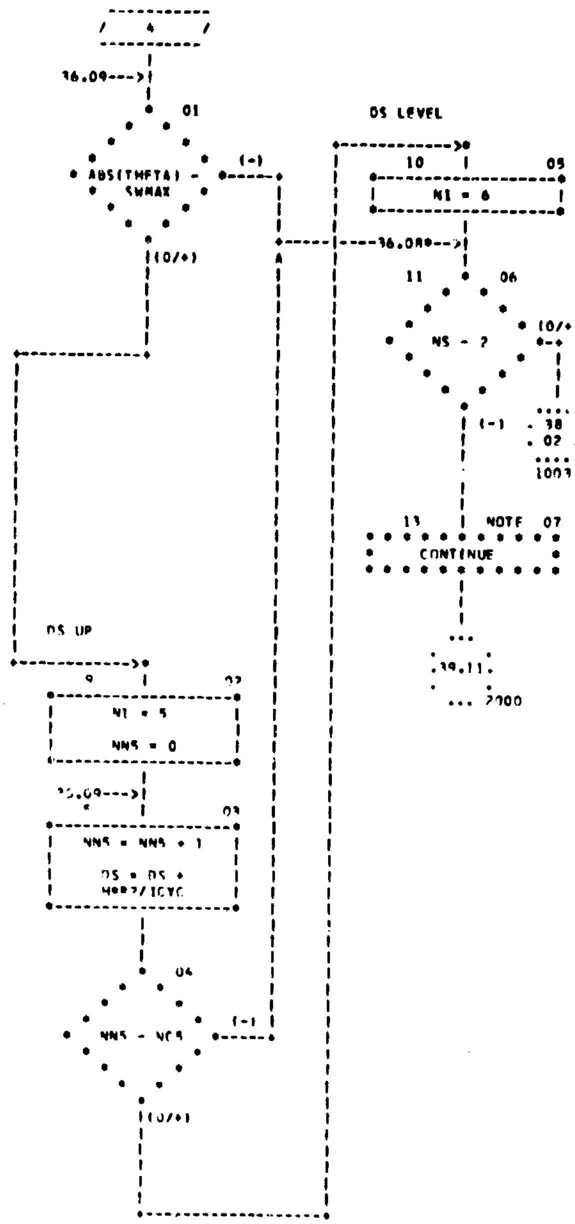


CHART TITLE - SUBROUTINE CONTR(THETA)

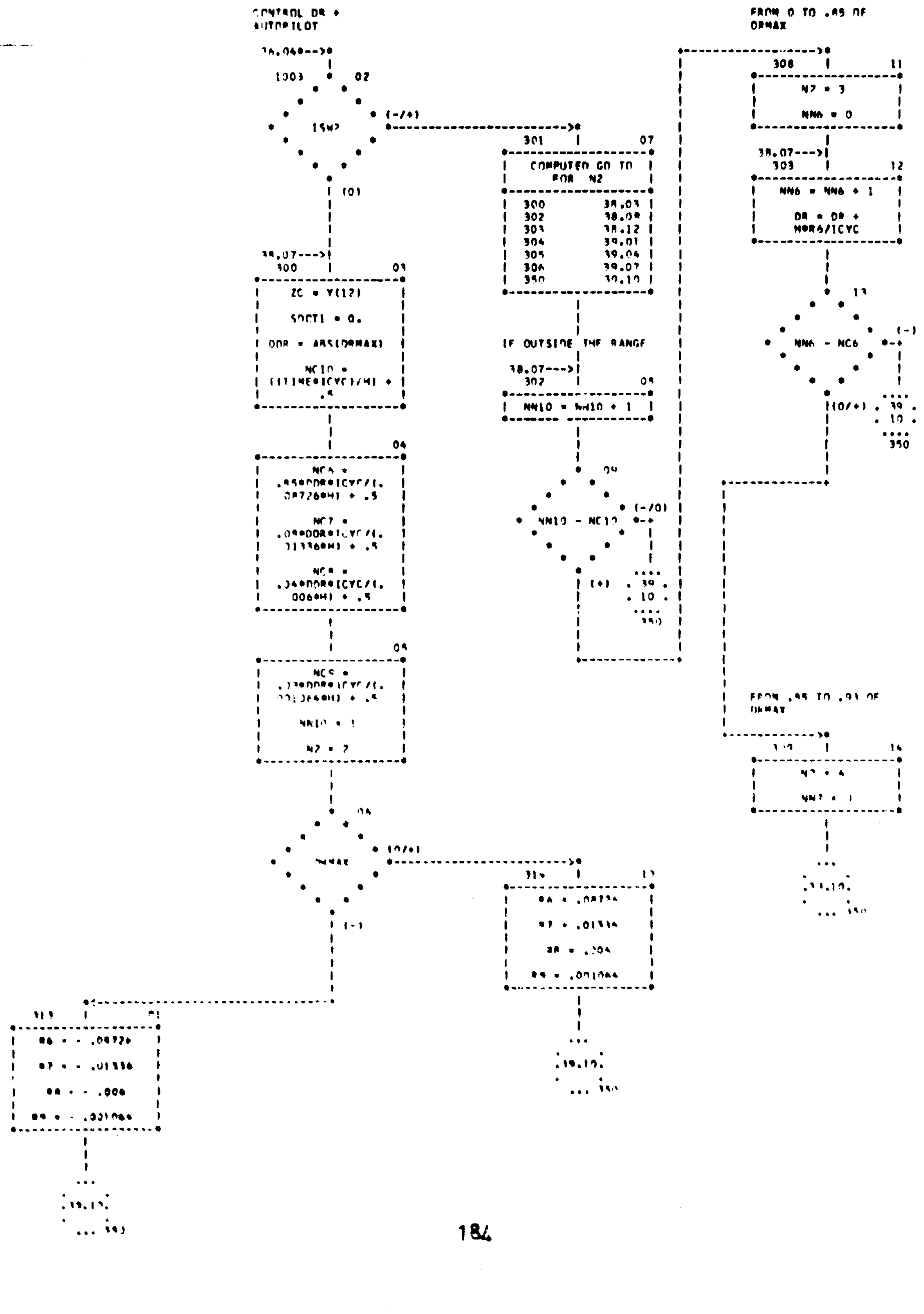


CHART TITLE - SUBROUTINE CONTRL(THETA)

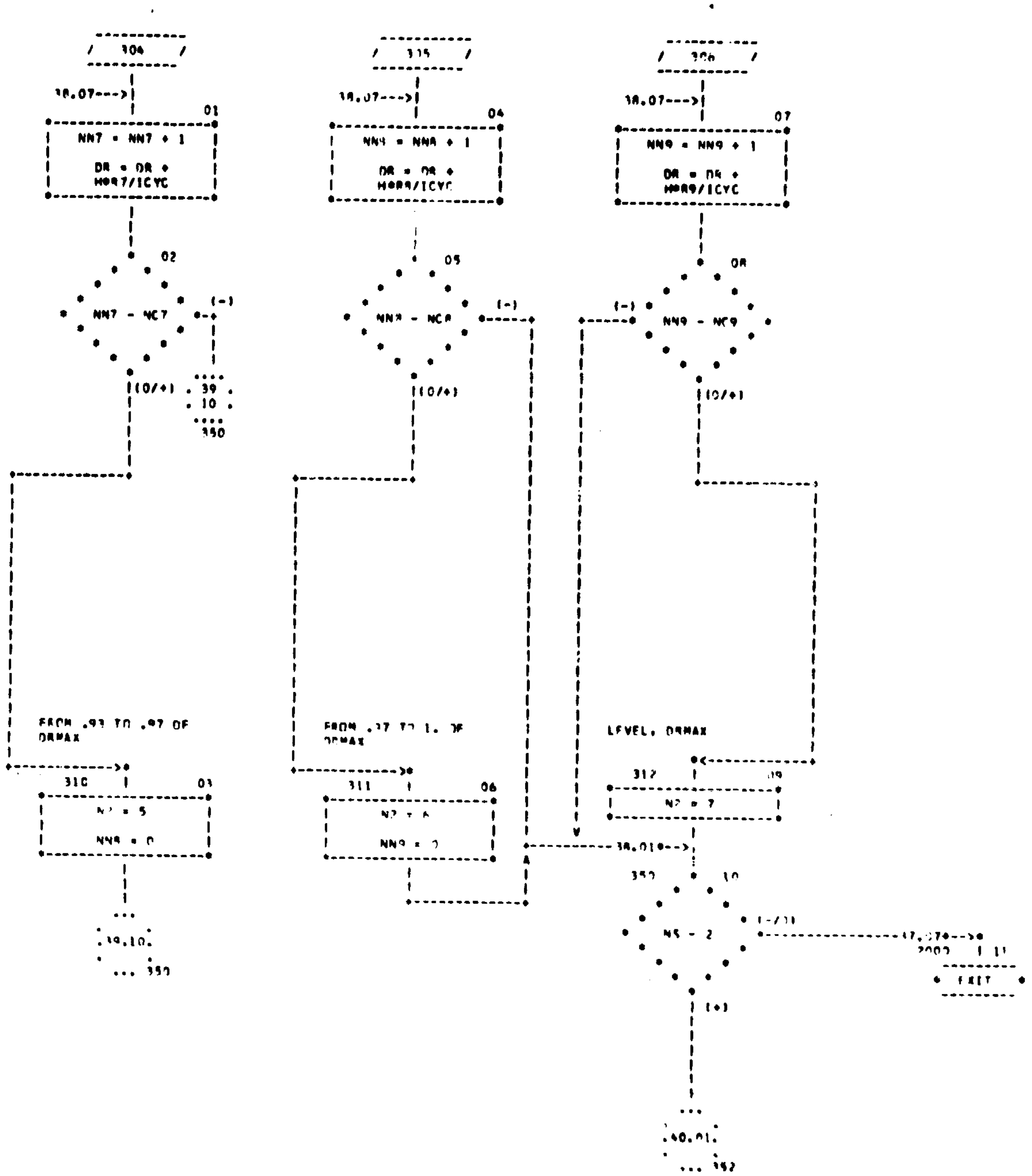


CHART TITLE - SUBROUTINE CONTR(THETA)

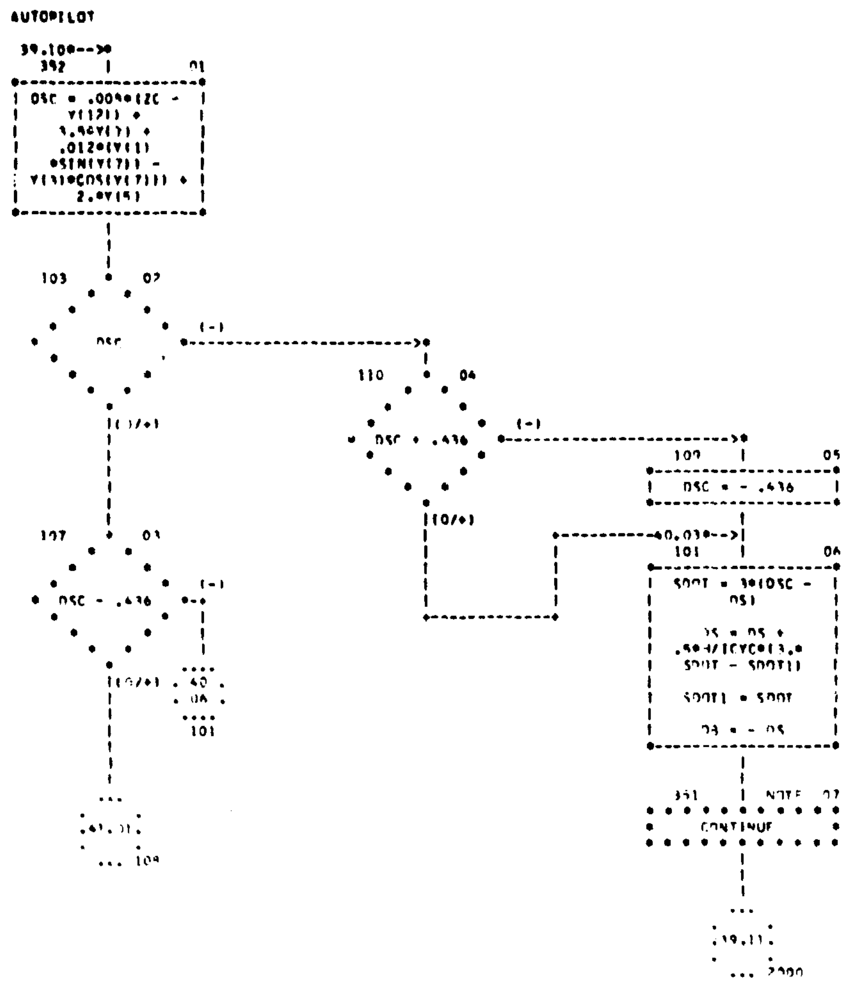
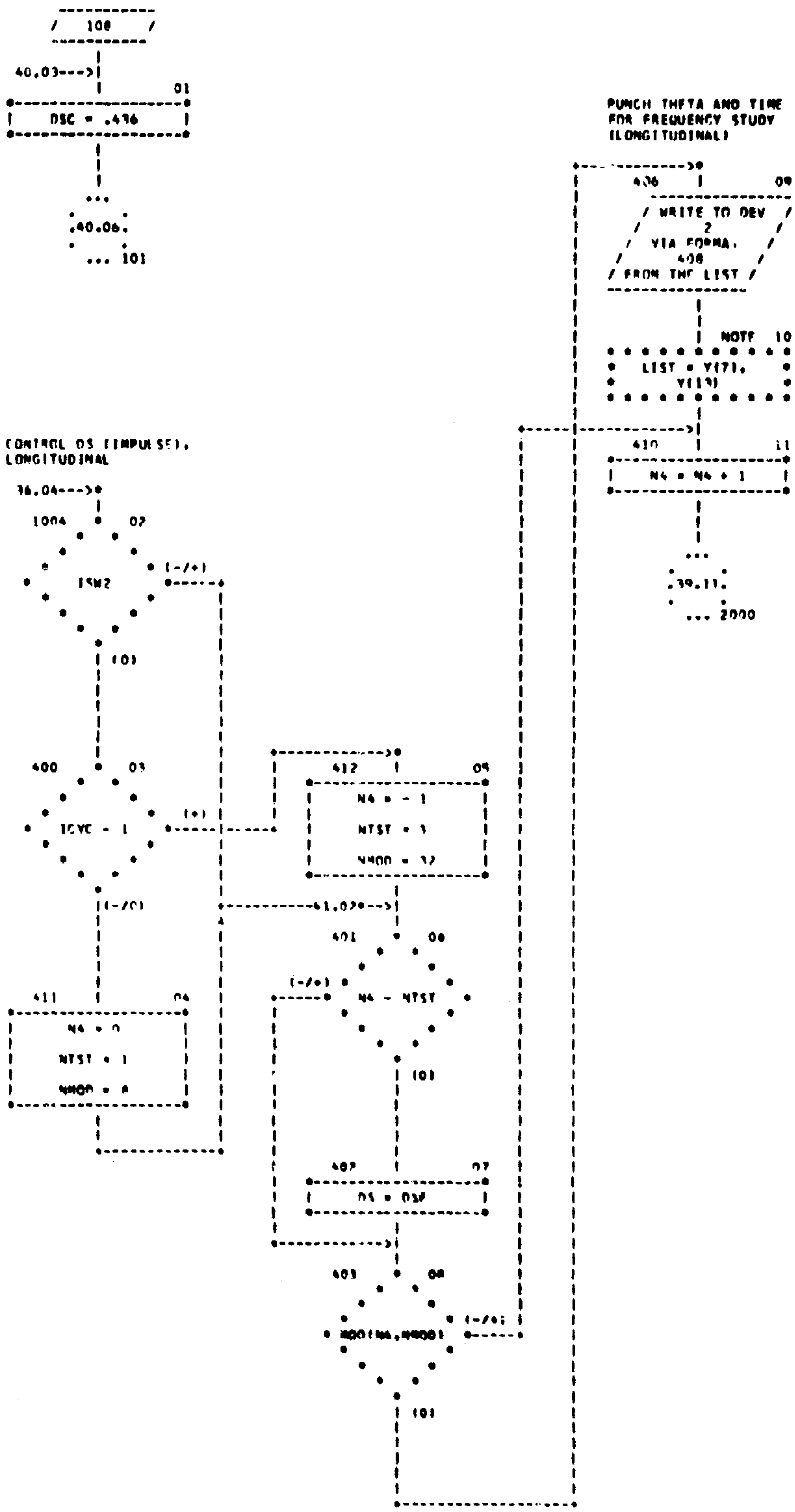


CHART TITLE - SUBROUTINE CONTR(THETA)



03/10/69 AUTOFLOW CHART SET - E8920 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE CONTR(THETA)

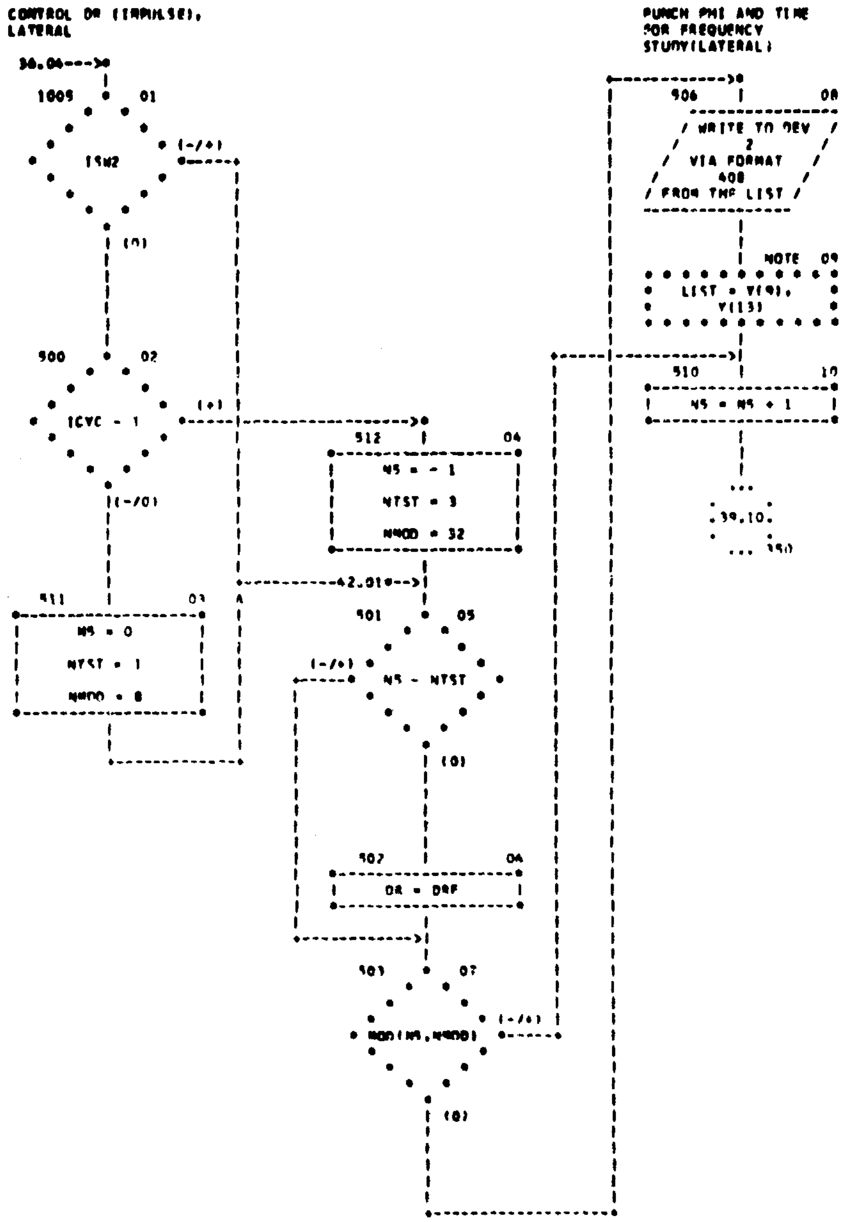
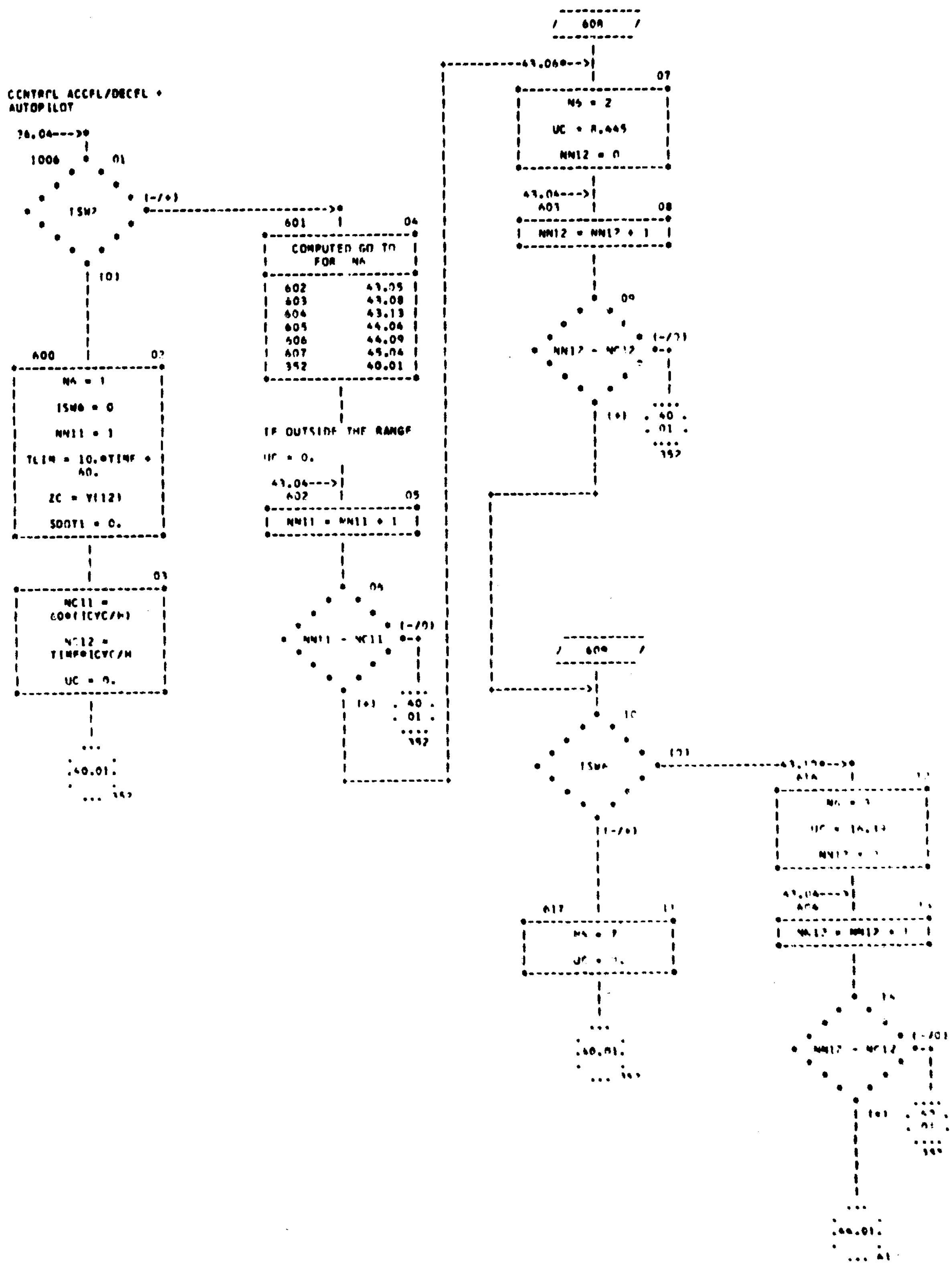


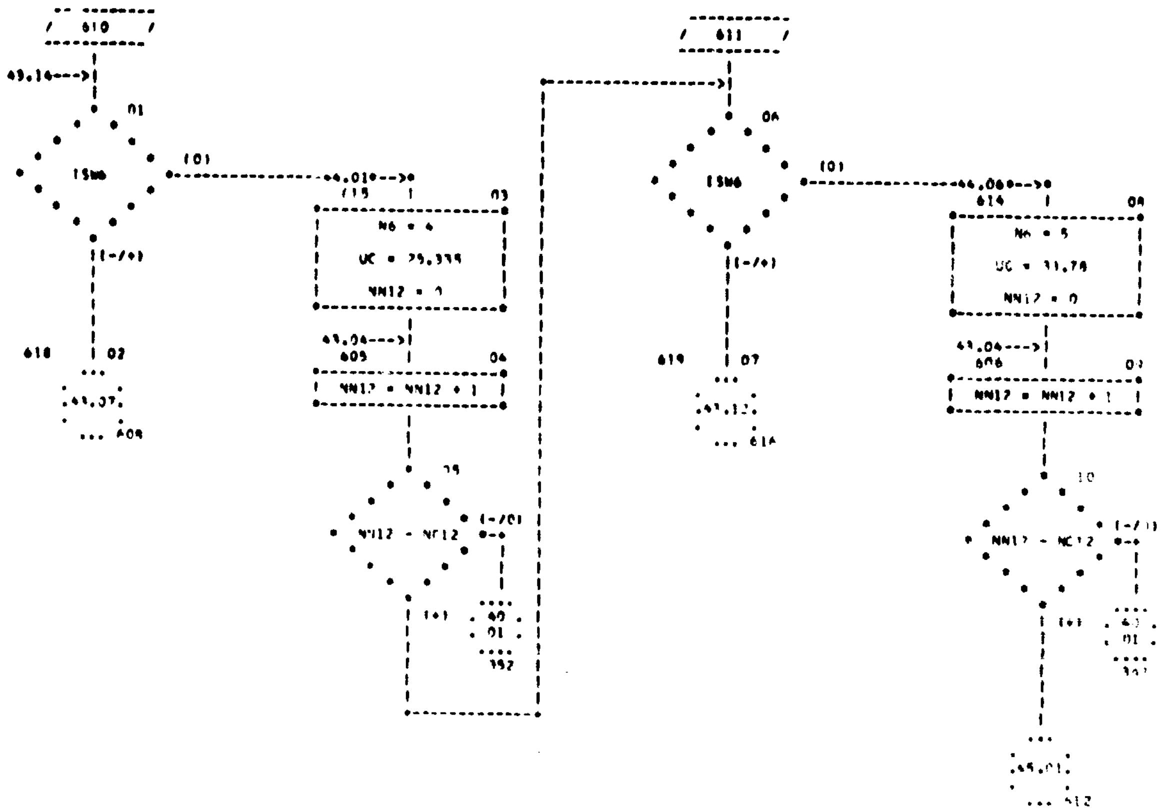
CHART TITLE - SUBROUTINE CONTR(THETA)



03/10/69

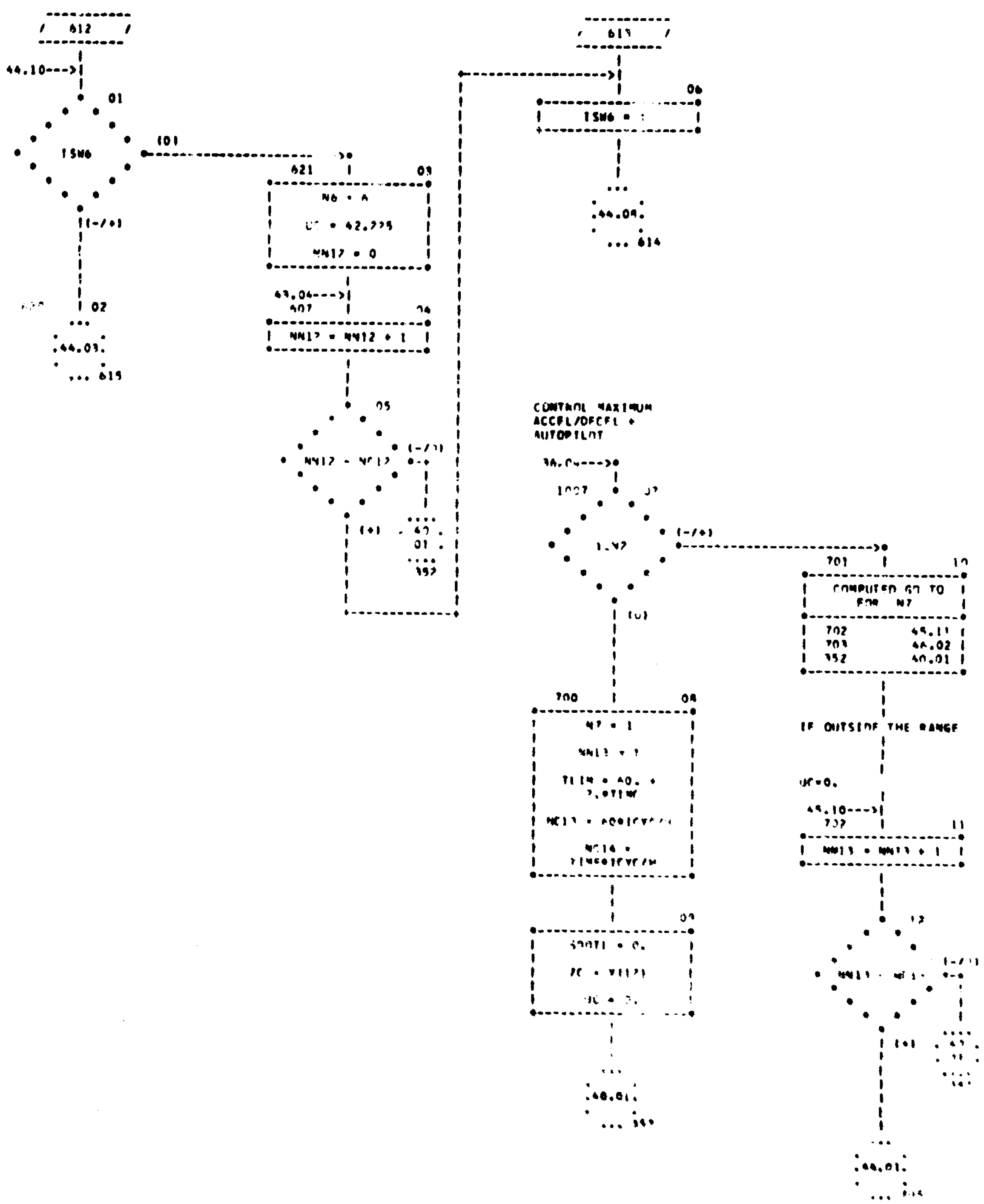
AUTOFLOW CHART SET - EF920 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE CONTR(ITHETA)



03/10/69 AUTOFLOW CHART SET - FR920 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE CONTRITHETA1



03/10/69 AUTOFLOW CHART SET - EB920 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE CONTR(THETA)

```
                        -----------
                       /   705    /
                       -----------
                            |
               45.12--->|
                            |             01
                *-------------------*
                |     N7 = 2        |
                |                   |
                |   UC = 42.225     |
                |                   |
                |    NN14 = 0       |
                *-------------------*
                            |
               45.10--->|
                703         |             02
                *-------------------*
                | NN14 = NN14 + 1   |
                *-------------------*
                            |
                            |
                            *    03
                        *       *
                     *             *  (-/0)
                  *  NN14 - NC14   *-*
                     *             *   |
                        *       *      |
                            *          ....
                            | (+)     . 40 .
                            |         . 01 .
                            |          ....
                            |           352
            *---------------*
            |
            |
            |
            |
            |           -----------
            |          /   706    /
            |          -----------
            |               |
            *------------->|
                            |             04
                *-------------------*
                |     N7 = 3        |
                |                   |
                |     UC = 0.       |
                *-------------------*
                            |
                            |
                           ...
                          .40.01.
                           ... 352
```

```
//      JOB    ZC790
//      EXEC FFORTRAN
      DIMENSION Y(6)
      DIMENSION SAVE(300, 8), ILOC( 8), BUFF(3000)
      REAL IY
C
      COMMON H,N,IS1,NPNT
C
      COMMON TLIM, RHOL2, RHOL3, RHOL4, RHOL5, WMB, ETA, ETAM1, ISW2
C
      COMMON         XQQ, XUD, XWQ, XUU, XWW, XDSDS, XDBDB, XWWE,
     1               XDSDSE
C
      COMMON         ZQD, ZWD, ZQ, ZAQDS, ZWAQ, ZSTR, ZW, ZWAW,
     1               ZAW, ZWW, ZDS, ZDB, ZQE, ZWE, ZWAWE, ZDSE
C
      COMMON         AMQD, AMQAQ, AMWD, AMQ, AMAQDS, AMAWQ, AMSTR, AMW,
     1               AMWAW, AMAW, AMWW, AMDS, AMDB, AMQE, AMWE, AMWAWE,
     2               AMDSE
C
      COMMON         IY
C
      COMMON         CW,CB,UC,XB,ZB
C
      COMMON         DS, DB, RHO, AL, AM
C
      COMMON         ETAHI, ETALO, A11, A12, A13
C
      COMMON         A21, A22, A23, A31, A32, A33
C
      COMMON         XG, ZG
C
      COMMON         ILOC, IPLOT, IRUN, IOPEN, NPLT,IOPT
C
      COMMON         Y
C
      COMMON TIME, R1, DELTMA, SWMAX, R2, DELTMI, DSE, ICYC, NS
C
      ICYC = 1
      PI = 3.141593
      IOPT = 0
      IOPEN = 0
      N = 5
   46 CONTINUE
      CALL INPUT
      IF(IPLOT)48,50,48
   48 IF(IOPEN) 50,49,50
   49 CALL PLOTS(BUFF, 12000, 7)
      IOPEN = 1
   50 CONTINUE
      NLOC=16
      K = 0
      IOUT = 3
      LINSPP=50
```

```
      LINS=99
      IPAGE=1
      WRITE(IOUT,24)IPAGE
      ICNT = NPNT
      ICNT2 = NPLT
C
C  COMPUTE RHO * L CONSTANTS
C
      RHOH = RHO * .5
      RHOL2 = RHOH * AL * AL
      RHOL3 = RHOL2 * AL
      RHOL4 = RHOL3 * AL
      RHOL5 = RHOL4 * AL
C
C  WRITE OUT HYDRODYNAMIC COEFFICIENTS
C
      WRITE(IOUT,1)XQQ,ZQD,AMQD,XUD,ZWD,AMQAQ,XWQ,ZQ,AMWD,XUU,ZAQDS,AMQ,
     1XWW,ZWAQ,AMAQDS,XDSDS,ZSTR,AMAWQ,XDBDB,ZW,AMSTR,XWWE,ZWAW,AMW,
     2XDSDSE,ZAW,AMWAW,ZWW,AMAW,ZDS,AMWW,ZDB,AMDS
    1 FORMAT(1H ,'XQQ',T9,E12.5,T44,'ZQD',T51,E12.5,T86,'MQD',T93,E12.5,
     1/1H ,'XUD',T9,E12.5,T44,'ZWD', T51,E12.5, T86,'MQAQ', T93, E12.5,/
     21H ,'XWQ',T9,E12.5, T44,'ZQ',  T51,E12.5, T86,'MWD',  T93, E12.5,/
     31H ,'XUU',T9,E12.5, T44,'ZAQDS',T51,E12.5,T86,'MQ',   T93, E12.5,/
     41H ,'XWW',T9,E12.5, T44,'ZWAQ',T51,E12.5, T86,'MAQDS',T93, E12.5,/
     51H ,'XDSDS',T9,E12.5,T44,'ZSTR',T51,E12.5,T86,'MAWQ', T93, E12.5,/
     61H ,'XDBDB',T9,E12.5,T44,'ZW', T51,E12.5, T86,'MSTR', T93, E12.5,/
     71H ,'XWWE',T9,E12.5,T44,'ZWAW',T51,E12.5, T86,'MW',   T93, E12.5,/
     81H ,'XDSDSE',T9,E12.5,T44,'ZAW',T51,E12.5,T86,'MWAW', T93, E12.5,/
     91H ,                   T44,'ZWW', T51,E12.5, T86,'MAW',  T93, E12.5,/
     A1H ,                   T44,'ZDS', T51,E12.5, T86,'MWW',  T93, E12.5,/
     B1H ,                   T44,'ZDB', T51,E12.5, T86,'MDS',  T93, E12.5)
      WRITE(IOUT,2)ZQE,AMDB,ZWE,AMQE,ZWAWE,AMWE,ZDSE,AMWAWE,AMDSE
    2 FORMAT(1H ,              T44,'ZQE', T51,E12.5, T86,'MDB',  T93, E12.5,/
     11H ,                   T44,'ZWE', T51,E12.5, T86,'MQE',  T93, E12.5,/
     21H ,                   T44,'ZWAWE',T51,E12.5,T86,'MWE',  T93, E12.5,/
     31H ,                   T44,'ZDSE',T51,E12.5, T86,'MWAWE',T93, E12.5,/
     41H ,                                         T86,'MDSE', T93,E12.5//)
      WRITE(IOUT,3)IY,CW,CB,UC,XB,ZB,DS,DB,RHO,AL,
     1AM, A11, A21, A31, ETAHI, ETALO, A12, A22,A32, XG, ZG, A13, A23,
     2A33, H
    3 FORMAT(1H ,                     T23,'IY',T30,E12.5/
     11H , 'W',     T9, E12.5,T23,'B',  T30,E12.5,  T44,'UC',   T51,E12.5,
     2T65, 'XB',    T72,E12.5,                       T107,'ZB', T114,E12.5/
     31H ,                   T23,'DS', T30,E12.5, T44, 'DB', T51, E12.5,
     4T65, 'RHO',   T72,E12.5,T86, 'L', T93,E12.5, T107, 'M', T114,E12.5/
     51H , 'A11',   T9, E12.5,T23,'A21',T30,E12.5, T44, 'A31',T51, E12.5,
     6                    T86,'ETAHI',T93,E12.5,T107,'ETALO',T114,E12.5/
     71H , 'A12',   T9, E12.5,T23,'A22',T30,E12.5, T44, 'A32',T51, E12.5,
     8T65, 'XG',    T72,E12.5,                       T107,'ZG', T114,E12.5/
     91H , 'A13',   T9, E12.5,T23,'A23',T30,E12.5, T44, 'A33',T51, E12.5,
     AT65, 'H',     T72,E12.5)
      WRITE(IOUT,4) TIME, R1, DELTMA, SWMAX, R2, DELTMI, DSE, ICYC, NS
    4 FORMAT(1H ,'TIME',T9,E12.5,T23,'R1',T30,E12.5,T44,'DELTMA',T51,E12
     1.5,T65,'SWMAX',T72,E12.5,T86,'R2',T93,E12.5,T107,'DELTMI',T114,E12
```

```
     2.5/1H ,'DSE', T9,F12.5, T44,'ICYC',T51,I2, T65,'NS', T72,I2/)
      WRITE(IOUT,5) IRUN
    5 FORMAT(1H , 'RUN NO  ', I5/)
C
C  COMPUTE W-B
C
      WMB=CW-CB
      IS1 = 0
      ISW2 = 0
   69 CALL INTEG(Y)
C
C  SAVE VALUES FOR PLOTS IF IPLOT = 1
C
      IF(IPLOT)9,12,9
    9 CONTINUE
C
C IF ARRAY FULL DONT OVERUN
C
      IF(K-300) 63,12,12
   63 CONTINUE
      IF(NPLT-ICNT2) 52,52,51
   52 ICNT2 = 0
      K = K + 1
C  SAVE TIME
      SAVE(K,1) = Y(6)
C  SAVE U
      SAVE(K,2) = Y(1)
C  SAVE W
      SAVE(K,3) = Y(2)
C  SAVE Q
      SAVE(K,4) = Y(3)
C  SAVE THETA
      SAVE(K,5) = Y(4)
C  SAVE Z
      SAVE(K,6) = Y(5)
C  SAVE DS
      SAVE(K,7) = DS
C  SAVE DB
      SAVE (K,8) = DB
   51 ICNT2 = ICNT2 + 1
   12 CONTINUE
      IF(NPNT-ICNT) 16,16,17
   16 ICNT = 0
      IF(LINSPP-LINS)20,20,30
   20 LINS=0
      IPAGE=IPAGE+1
      WRITE(IOUT,24)IPAGE
   24 FORMAT(1H1,7X,'ZC790',20X,'SUBMARINE SIMULATION, LONGITUDINAL FREE
     1DOM',45X,'PAGE',I8//)
      WRITE(IOUT,25)
   25 FORMAT(1H ,9X,'U',16X,'W',16X,'Q',14X,'THETA',14X,'Z',15X,'T')
   30 CONTINUE
   72 WRITE(IOUT,10) (Y(I),I=1,6)
   10 FORMAT(1H0,6(2X,E13.6,2X))
```

```
74 CONTINUE
   LINS=LINS+2
17 ICNT = ICNT + 1
   IF(ABS(Y(6)-TLIM)-H)35,69,69
35 CONTINUE
   IF(IPLOT) 40,46,40
40 CALL PLTROU(SAVE,K,ILOC,NLOC,IRUN)
   GO TO 46
   END
```

```
      SUBROUTINE INPUT
      DIMENSION Y(6), ILOC(8), YHOLD(6), COM(219)
      EQUIVALENCE (COM(1),H)
      REAL IY
C
      COMMON H, N, IS1, NPNT
C
      COMMON TLIM, RHOL2, RHOL3, RHOL4, RHOL5, WMB, ETA, ETAM1, ISW2
C
      COMMON        XQQ, XUD, XWQ, XUU, XWW, XDSDS, XDBDB, XWWE,
     1              XDSDSE
C
      COMMON        ZQD, ZWD, ZQ, ZAQDS, ZWAQ, ZSTR, ZW, ZWAW,
     1              ZAW, ZWW, ZDS, ZDB, ZQE, ZWE, ZWAWE, ZDSE
C
      COMMON        AMQD, AMQAQ, AMWD, AMQ, AMAQDS, AMAWQ, AMSTR, AMW,
     1              AMWAW, AMAW, AMWW, AMDS, AMDB, AMQE, AMWE, AMWAWE,
     2              AMDSE
      COMMON        IY
C
      COMMON        CW, CB, UC, XB, ZB
C
      COMMON        DS, DB, RHO, AL, AM
C
      COMMON        ETAHI, ETALO, A11, A12, A13
C
      COMMON        A21, A22, A23, A31, A32, A33
C
      COMMON        XG, ZG
C
      COMMON        ILOC, IPLOT, IRUN, IOPEN, NPLT,IOPT
C
      COMMON        Y
C
      COMMON TIME, R1, DELTMA, SWMAX, R2, DELTMI, DSE, ICYC, NS
C
      IN   = 1
      IF(IOPT)150,5,150
    5 CONTINUE
      READ(IN,50) NPNT, IPLOT, IRUN, NPLT, IOPT, NS
      IF(IRUN)70,60,70
   60 IF(IOPEN) 62,64,62
   62 CALL PLOT(5.0,0.0,999)
   64 CONTINUE
      CALL EXIT
   70 CONTINUE
      READ(IN,50) (ILOC(I), I = 1,8)
   50 FORMAT(16I5)
      READ(IN,100)TO,HO,TLIM
      H = HO
  100 FORMAT(9F12.5)
      READ(IN,100)(Y(I), I=1,5)
      Y(6) = TO
C
```

```
      READ(IN,100) XQQ, XUD, XWQ, XUU, XWW, XDSDS, XDBDB, XWWE,
     1 XDSDSE
C
      READ(IN,100) ZQD, ZWD, ZQ, ZAQDS, ZWAQ, ZSTR, ZW, ZWAW,
     1             ZAW, ZWW, ZDS, ZDB, ZQE, ZWE, ZWAWE, ZDSE
C
      READ(IN,100) AMQD, AMQAQ, AMWD, AMQ, AMAQDS, AMAWQ, AMSTR, AMW,
     1             AMWAW, AMAW, AMWW, AMDS, AMDB, AMQE, AMWE, AMWAWE,
     2             AMDSE
C
C
      READ(IN,100) IY
C
      READ(IN,100) CW, CB, UC, XB, ZB
C
      READ(IN,100) DS, DB, RHO, AL, AM
C
      READ(IN,100) ETAHI, ETALO, A11, A12, A13
C
      READ(IN,100)  A21, A22, A23, A31, A32, A33
C
      READ(IN,100) XG, ZG
C
      READ(IN,100) TIME, R1, DELTMA, SWMAX, R2, DELTMI, DSF
C
C  SAVE  INITIAL  CONDITIONS  FOR  POSSIBLE  RESTORE
C
      DO 110 I=1,6
      YHOLD(I)= Y(I)
  110 CONTINUE
C
C  SAVE  DS,DB FOR POSSIBLE RESTORE
C
      DSHOLD = DS
      DBHOLD = DB
      RETURN
  150 CONTINUE
C
C  RESTORE  INITIAL VALUES
C
      DO 152 I=1,6
      Y(I) = YHOLD(I)
  152 CONTINUE
C
C  RESTORE INITIAL DS,DB,H
C
      DS =  DSHOLD
      DB = DBHOLD
      H  =  HO
      READ(IN,165) IRUN
      IF(IRUN) 155,05,155
  155 CONTINUE
  160 READ(IN,165) NDEX, VALUE
  165 FORMAT(I5,5X,F10.5)
```

```
      IF(NDEX)180,170,180
  170 RETURN
  180 COM(NDEX) = VALUE
      GO TO 160
      END
```

```
      SUBROUTINE INTFG(Y)
      DIMENSION Y(1), F(5), F1(5)
      COMMON H, N, IS1
      IF(IS1)30,10,30
   10 M= N + 1
      DO 20 I = 1,N
      F(I) = 0.
   20 F1(I) = 0.
      IS1 = 1
      RETURN
   30 CALL EVAL1(Y,F)
      DO 40 I = 1,N
C
      Y(I) = Y(I) + .5*H*(3.*F(I)-F1(I))
C
      F1(I) = F(I)
   40 CONTINUF
      Y(M) = Y(M)+H
      RETURN
      END
```

```
      SUBROUTINE EVAL1(YI,F)
      DIMENSION YI(1),F(1)
      REAL IY
C
      COMMON H, N, IS1, NPNT
C
      COMMON TLIM, RHOL2, RHOL3, RHOL4, RHOL5, WMB, ETA, ETAM1, ISW2
C
      COMMON       XQQ, XUD, XWQ, XUU, XWW, XDSDS, XDBDB, XWWE,
     1             XDSDSE
C
      COMMON       ZQD, ZWD, ZQ, ZAQDS, ZWAQ, ZSTR, ZW, ZWAW,
     1             ZAW, ZWW, ZDS, ZDB, ZQE, ZWE, ZWAWE, ZDSE
C
      COMMON       AMQD, AMQAQ, AMWD, AMQ, AMAQDS, AMAWQ, AMSTR, AMW,
     1             AMWAW, AMAW, AMWW, AMDS, AMDB, AMQE, AMWE, AMWAWE,
     2             AMDSE
C
      COMMON       IY
C
      COMMON       CW, CB, UC, XB, ZB
C
      COMMON       DS, DB, RHO, AL, AM
C
      COMMON       ETAHI, ETALO, A11, A12, A13
C
      COMMON       A21, A22, A23, A31, A32, A33
C
      COMMON       XG, ZG
C
C
C  PULL PRESENT VALUES OF VARIABLES OUT OF ARRAY YI
C
      U = YI(1)
      W = YI(2)
      Q = YI(3)
      THETA=YI(4)
      Z = YI(5)
      CALL CONTR(THETA)
C
C COMPUTE QUANTITIES TO BE USED MORE THAN ONCE
C
      Q2 = Q*Q
      U2 = U*U
      W2 = W*W
      DS2 = DS*DS
      DS2U2 = DS2*U2
      WQ = W*Q
      ROOTVW=SQRT(W2)
      UQ=U*Q
      ABSQ=ABS(Q)
      UAQDS=U*ABSQ*DS
      WRTVW=W*ROOTVW
      UW=U*W
```

```
      ABSW=ABS(W)
      UABSW=U*ABSW
      DBU2=DB*U2
      DSU2=DS*U2
      UMAG = SQRT(U2+W2)
      IF(UMAG)26,24,26
   24 ETA = 20.
      GO TO 28
   26 CONTINUE
      ETA = UC/UMAG
   28 ETAM1 = ETA-1.
      IF(ETA-ETAHI)32,30,30
   30 A1=A11
      A2=A12
      A3=A13
      GO TO 38
   32 IF(ETA-ETALO)35,35,37
   35 A1=A31
      A2=A32
      A3=A33
      GO TO 38
   37 A1=A21
      A2=A22
      A3=A23
   38 CONTINUE
    1 RATVAV = 0.
    3 IF(W) 5,4,5
    4 RATWAW = 0.
      GO TO 6
    5 RATWAW = W/ABSW
    6 CONTINUE
C
C  COMPUTE TRIG FUNCTIONS
C
      SPHI=0.
      CPHI=1.
      STTA = SIN(THETA)
      CTTA = COS(THETA)
      SPSI=0.
      CPSI=1.
      TRIG1=0.
      TRIG2=CTTA
      TRIG3=0.
      TRIG4=STTA
      TRIG5 = U*CTTA
      IF(ISW2)20,10,20
   10 ISW2 = 1
C
C  SET COEFFICIENTS OF UD, WD, QD
C
      FAU    = AM-RHOL3*XUD
      FAQ    = AM*ZG
      FNW     = AM-RHOL3*ZWD
      FNQ     = -RHOL4*ZQD-AM*XG
```

```
      PMU     = AM*ZG
      PMW     = -RHOL4*AMWD-AM*XG
      PMQ     = IY-RHOL5*AMQD
C
   20 CONTINUE
C
C  COMPUTE UD FROM AXIAL FORCE EQN
C
   40 CONTINUE
      F(1)=((AM*(-WQ) + RHOL4*(XQQ*Q2) +
     1RHOL3*( XWQ*WQ)+RHOL2*(XUU*U2+XWW*W*W)+
     2RHOL2*U2*(+XDSDS*DS2+XDBDB*DB*DB) +
     3RHOL2*(A1*U2+A2*U*UC+A3*UC*UC)-WMB*STTA+ RHOL2*(XWWE*W2+
     4XDSDSE*DS2U2)*ETAM1) + AM*XG*Q2
     5-FAQ* F(3))/FAU
C
C  COMPUTE WD FROM NORMAL FORCE EQN
C
   60 CONTINUE
      F(2)=(AM*UQ + RHOL3*(
     1ZQ*UQ +ZAQDS* UAQDS + ZWAW* RATWAW*ROOTVW*ABSQ) +
     2RHOL2 *(ZSTR*U2 + ZW*UW+ZWAW * WRTVW +ZAW*UABSW+ZWW*ABSW*ROOTVW
     3 +ZDS*DSU2+ZDB*DBU2) + WMB * TRIG2 + RHOL3* ZQE*UQ* ETAM1
     4+RHOL2*(ZWE*UW+ZWAWE*WRTVW+ZDSE*DSU2)*ETAM1
     5 +AM*ZG*Q2-FNQ*F(3))/FNW
C
C  COMPUTE QD FROM PITCHING MOMENT EQN
C
   80 CONTINUE
      F(3)=(RHOL5*(AMQAQ*Q*ABSQ) + RHOL4*(AMQ*UQ +AMAQDS*
     1UAQDS +AMAWQ *Q*ROOTVW)+ RHOL3*(AMSTR*U2+AMW*UW+AMWAW*WRTVW +
     2AMAW* UABSW +AMWW*ABSW*ROOTVW+AMDS*DSU2+AMDB*DBU2)
     3-(XG*CW-XB*CB)*TRIG2-(ZG*CW-ZB*CB)*STTA+RHOL4*AMQE*UQ*ETAM1
     4+RHOL3*(AMWE*UW+AMWAWE*WRTVW+AMDSE*DSU2)*ETAM1
     5 +AM*(ZG*(-WQ)+XG*(-UQ))
     6 -PMU*F(1)-PMW*F(2))/PMQ
C
C  COMPUTE KINEMATICS - THETA DOT
C
      F(4) = Q
C
C  COMPUTE Z DOT
C
      F(5)=-U*STTA+W*TRIG2
      RETURN
      END
```

```
      SUBROUTINE PLTRDU(SAVE,K,ILOC,NLOC,IRUN)
      DIMENSION  SAVE(300,1),ILOC(1), IY(16), IR(2)
      IR(1)=IHEX(13,9,14,4,13,5,4,0)
      IR(2)=IHEX(13,5,13,6,4,11,4,0)
C   T
      IY(1) = IHEX(14,3,4,0,4,0,4,0)
C   U
      IY(2)=IHEX(14,4,4,0,4,0,4,0)
C   W
      IY(3)=IHEX(14,6,4,0,4,0,4,0)
C   Q
      IY(4)=IHEX(13,8,4,0,4,0,4,0)
C   THTA
      IY(5)=IHEX(14,3,12,8,14,3,12,1)
C   Z
      IY( 6)=IHEX(14,9,4,0,4,0,4,0)
C   DS
      IY( 7)=IHEX(12,4,14,2,4,0,4,0)
C   DB
      IY( 8)=IHEX(12,4,12,2,4,0,4,0)
      DIV = 20.
      CALL SCALE(SAVE(1,1),6.0,K,1,DIV,1)
      ICTL=0
      CALL PLOT(0.0,.75,23)
      DO 80 I=1,NLOC
      J=ILOC(I)
      IF(J) 30,90,30
   30 CONTINUE
      CALL SCALE(SAVE(1,J),4.0,K,1,DIV,2)
      CALL AXIS(0.0,0.0,IY(1),-4,6.0,0.0,DIV,1)
      CALL AXIS(0.0,0.0,IY(J ),4,4.0,90.0,DIV,2)
      CALL SYMBOL(4.0,3.5,0.14,IR,0.0,8)
      AIRUN = IRUN
      CALL NUMBER(-0.0,-0.0,-0.0,AIRUN,0.0,-1)
      CALL LINE(SAVE(1,1),SAVE(1,J),K,1,0,0)
      IF(ICTL)60,50,60
   50 CALL PLOT(0.0,4.50,-23)
      ICTL=1
      GO TO 80
   60 CALL PLOT(8.5,-4.50,-23)
      ICTL=0
   80 CONTINUE
   90 IF(ICTL)110,100,110
  100 CALL PLOT(0.0,-.75,23)
      RETURN
  110 CALL PLOT(8.5,-5.25,23)
      RETURN
      END
```

```
      SUBROUTINE CONTR(THETA)
C
C TO CONTROL DS          FOR DYNAMIC CONDITIONS
C
      DIMENSION ILCC(8), Y(6)
      REAL IY
      REAL K
C
      COMMON H, N, IS1, NPNT
C
      COMMON TLIM, RHOL2, RHOL3, RHOL4, RHOL5, WMB, ETA, ETAM1, ISW2
C
      COMMON         XQQ, XUD, XWQ, XUU, XWW, XDSDS, XDBDB, XWWE,
     1               XDSDSF
C
      COMMON         ZQD, ZWD, ZQ, ZAQDS, ZWAQ, ZSTR, ZW, ZWAW,
     1               ZAW, ZWW, ZDS, ZDB, ZQE, ZWF, ZWAWF, ZDSF
C
      COMMON  AMQD,         AMQAQ, AMWD, AMQ, AMAQDS, AMAWQ, AMSTR, AMW,
     1               AMWAW, AMAW, AMWW, AMDS, AMDB, AMQE, AMWF, AMWAWE,
     2               AMDSE
C
C
      COMMON         IY
C
      COMMON         CW, CB, UC, XB, ZB
C
      COMMON         DS, DB, RHO, AL, AM
C
      COMMON         ETAHI, ETALO, A11, A12, A13
C
      COMMON         A21, A22, A23, A31, A32, A33
C
      COMMON XG, ZG
C
      COMMON         ILOC, IPLOT, IRUN, IOPEN, NPLT, IOPT
C
      COMMON         Y, TIME, R1, DELTMA, SWMAX, R2, DELTMI
C
      COMMON DSF, ICYC, NS
      IF(NS)15,15,16
   15 RETURN
   16 CONTINUE
      GO TO(1001,1002,1003,1004),NS
C
C CONTROL DS
C
 1001 IF(ISW2)21,20,21
   20 N1 = 2
    1 NN2 = 1
      NC2 = ((TIME*ICYC)/H) + .5
      NC3 = (ABS(DS      -DELTMA))*ICYC/ABS(R1*H) +.5
      NC5 = (ABS(DELTMI-DELTMA))*ICYC/ABS(R2*H) +.5
      GO TO 11
```

```
   21 GO TO (1,2,3,4,5,11),N1
C
C CYCLES TO START
C
    2 NN2 = NN2 + 1
      IF (NN2 - NC2) 11,11,7
C
C DS DOWN
C
    7 N1 = 3
      NN3 = 0
    3 NN3 = NN3 +1
      DS = DS   + H*R1/ICYC
      IF(NN3 - NC3) 11,08,8
C
C DS LEVEL
C
    8 N1 = 4
      GO TO 11
    4 IF (ABS(THETA) - SWMAX) 11,9,9
C
C DS UP
C
    9 N1 = 5
      NN5 = C
    5 NN5 = NN5 + 1
      DS = DS   + H*R2/ICYC
      IF (NN5 -NC5) 11,10,10
C
C DS LEVEL
C
   10 N1 = 6
   11 CONTINUE
      GO TO 2000
C
C AUTOPILOT
C
  352 DSC=.008*(ZC-Y(5))+3.5*Y(4)+.012*(Y(1)*SIN(Y(4))-Y(2)*COS(Y(4)))
     1+2.*Y(3)
  103 IF (DSC) 110,107,107
  107 IF (DSC - .436) 101,108,108
  110 IF (DSC + .436) 109,101,101
  108 DSC = .436
      GC TO 101
  109 DSC = -.436
  101 SDOT = 3 * (DSC -DS)
      DS = DS + .5 * H/ICYC * (3. * SDOT - SDOT1)
      SDOT1 = SDOT
      DB = -DS
  351 CONTINUE
      GO TO 2000
C
C CONTROL DS (IMPULSE), LONGITUDINAL
C
```

```
 1002 IF (ISW2)401,400,401
  400 N4 = 0
  401 IF (N4-1)403,402,403
  402 DS = DSF
  403 IF(MOD(N4, 8))410,406,410
C
C PUNCH THETA AND TIME FOR FREQUENCY STUDY (LONGITUDINAL)
C
  406 WRITE(2,408)Y(4),Y(6)
  408 FORMAT(2E15.7)
  410 N4 = N4 + 1
      GO TO 2000
C
C  CONTROL ACCEL/DECEL + AUTOPILOT
C
 1003 IF(ISW2)601,600,601
  600 N6=1
      ISW6=0
      NN11=1
      TLIM=     10.*TIME+60.
      ZC = Y( 5)
      SDOT1 = 0.
      NC11=60*(ICYC/H)
      NC12=TIME*ICYC/H
      UC=0.
      GO TO 352
  601 GO TO(602,603,604,605,606,607,352),N6
C
      UC=0.
  602 NN11=NN11+1
      IF(NN11-NC11)352,352,608
C
  608 N6=2
      UC=8.445
      NN12=0
  603 NN12=NN12+1
      IF(NN12-NC12)352,352,609
C
  609 IF(ISW6)617,616,617
  617 N6=7
      UC=0.
      GO TO 352
C
  616 N6=3
      UC=16.99
      NN12=0
  604 NN12=NN12+1
      IF(NN12-NC12)352,352,610
C
  610 IF(ISW6)618,615,618
  618 GO TO 608
C
  615 N6=4
      UC=25.335
```

```
      NN12=0
  605 NN12=NN12+1
      IF(NN12-NC12)352,352,611
C
  611 IF(ISW6)619,614,619
  619 GO TO 616
C
  614 N6=5
      UC=33.78
      NN12=0
  606 NN12=NN12+1
      IF(NN12-NC12)352,352,612
C
  612 IF(ISW6)620,621,620
  620 GO TO 615
C
  621 N6=6
      UC=42.225
      NN12=0
  607 NN12=NN12+1
      IF(NN12-NC12)352,352,613
C
  613 ISW6 = 1
      GO TO 614
C
C       CONTROL MAXIMUM ACCEL/DECEL + AUTOPILOT
C
 1004 IF(ISW2)701,700,701
  700 N7=1
      NN13=1
      TLIM=60.+2.*TIME
      NC13=60*ICYC/H
      NC14=TIME*ICYC/H
      SDOT1=0.
      ZC=Y( 5)
      UC=0.
      GO TO 352
  701 GO TO(702,703,352),N7
C
C     UC=0.
  702 NN13=NN13+1
      IF(NN13-NC13)352,352,705
C
  705 N7=2
      UC=42.225
      NN14=0
  703 NN14=NN14+1
      IF(NN14-NC14)352,352,706
C
  706 N7=3
      UC=0.
      GO TO 352
 2000 RETURN
      END
```

03/11/69

AUTOFLOW CHART SET - ZC790 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES

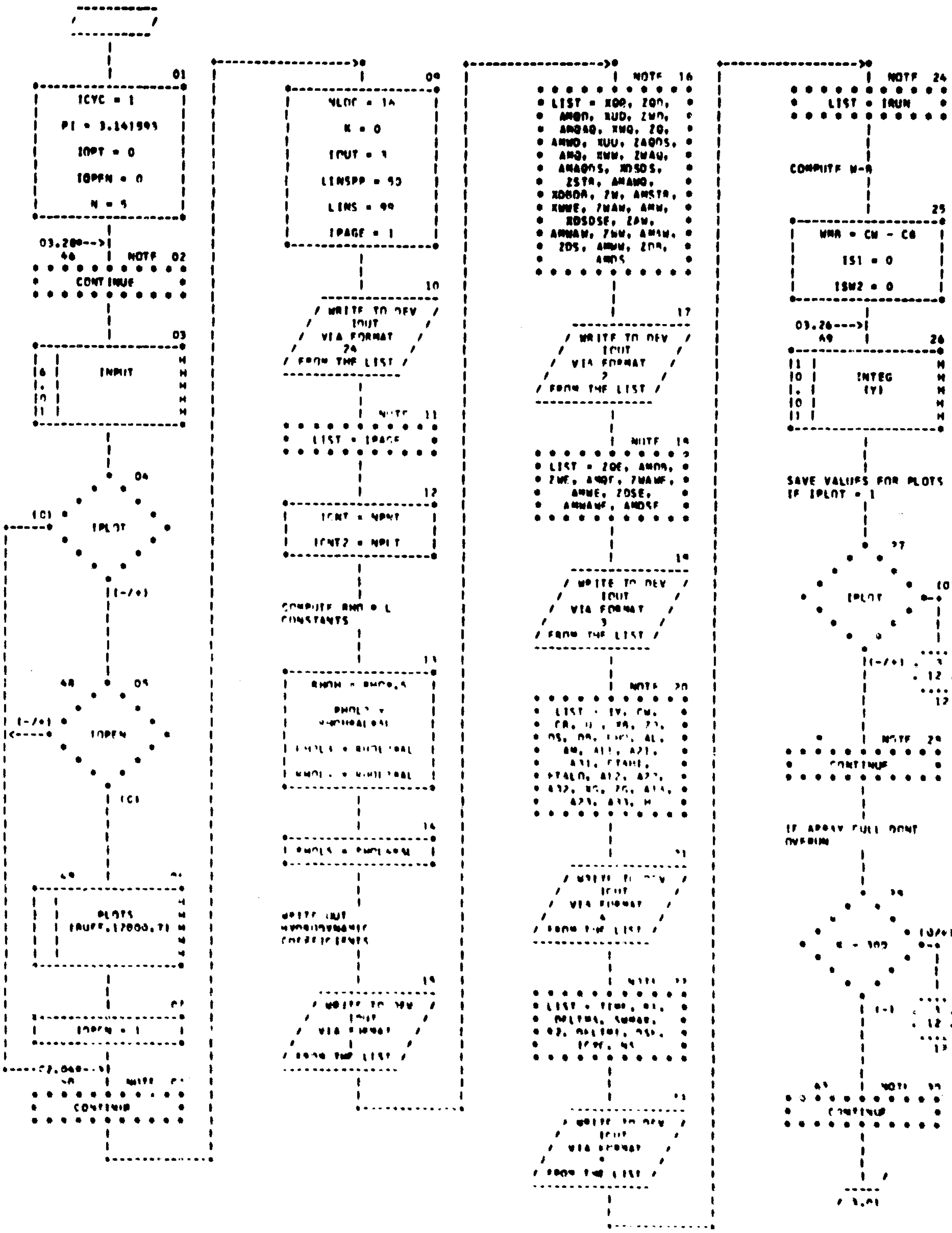
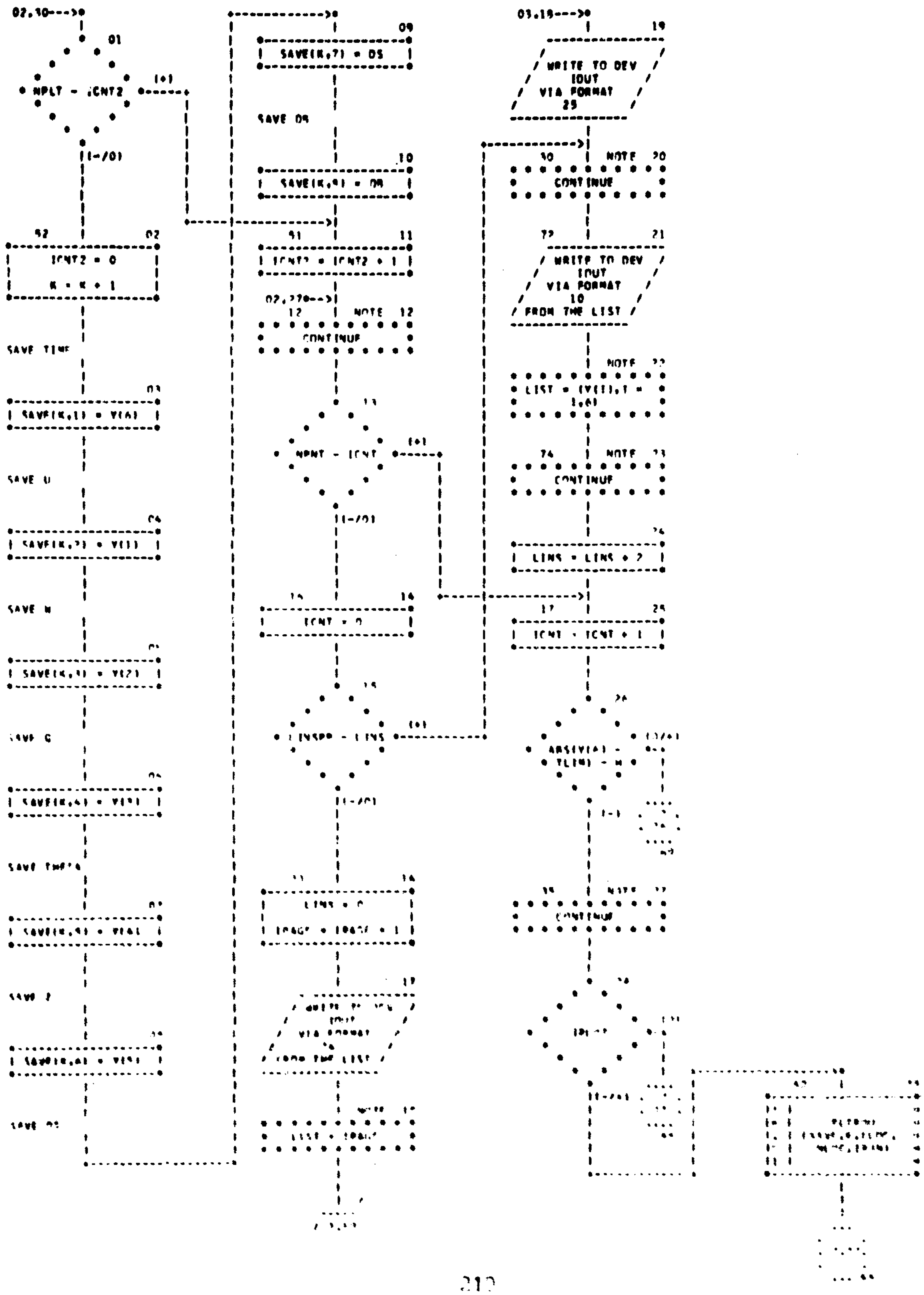


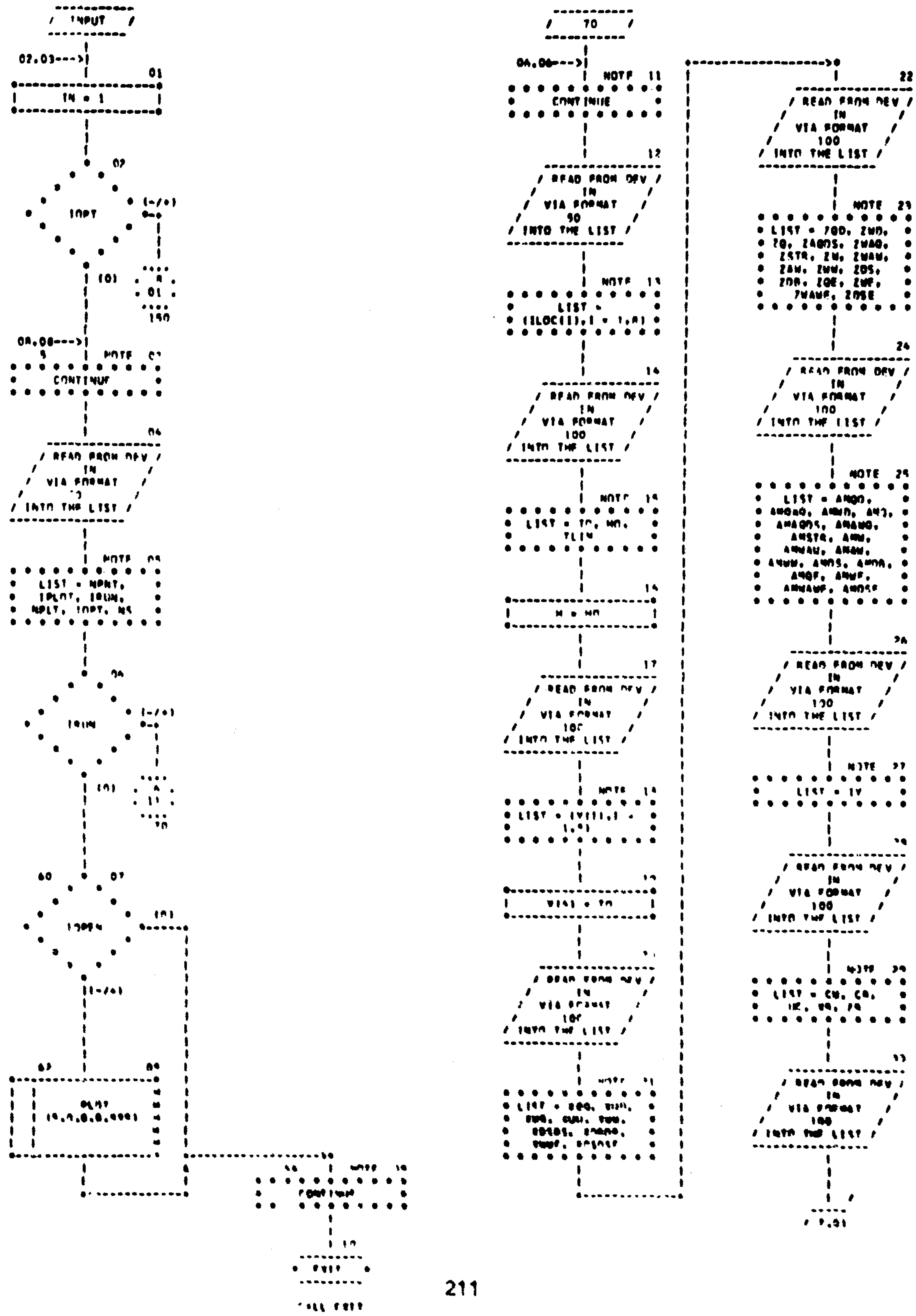
CHART TITLE - PROCEDURES



03/11/69

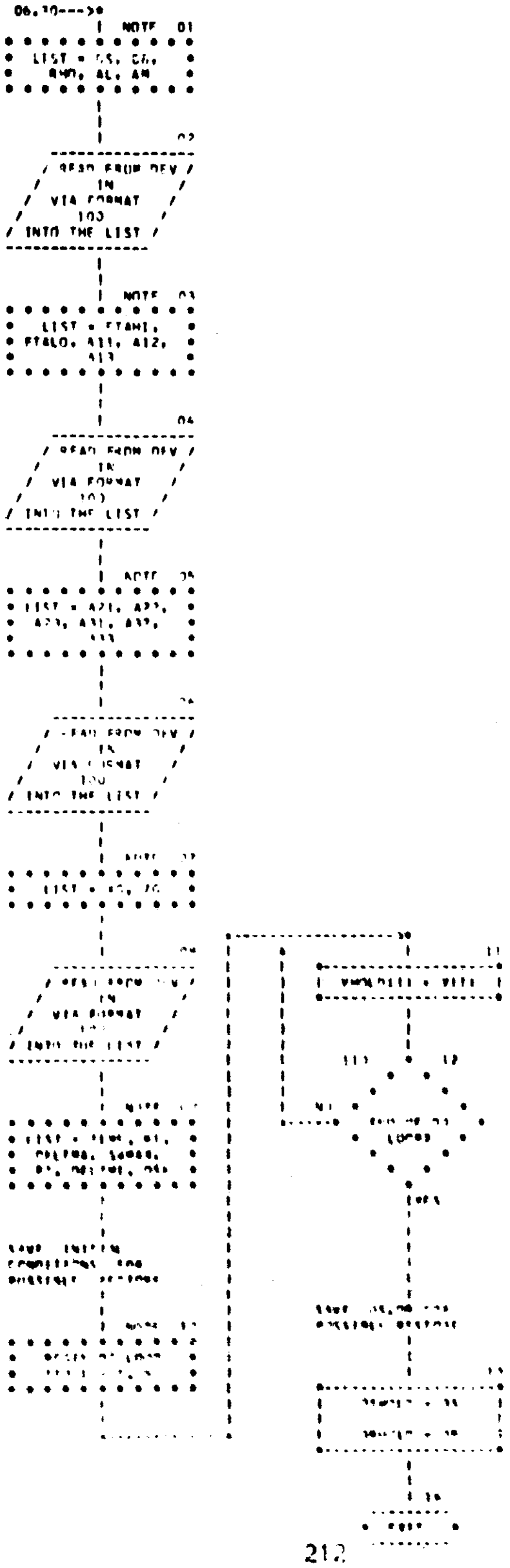
AUTOFLOW CHART SET - ZC790 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE INPUT



03/11/69 AUTOFLOW CHART SET - ZC790 NAVTRADEVCEN 68-C-0050-2

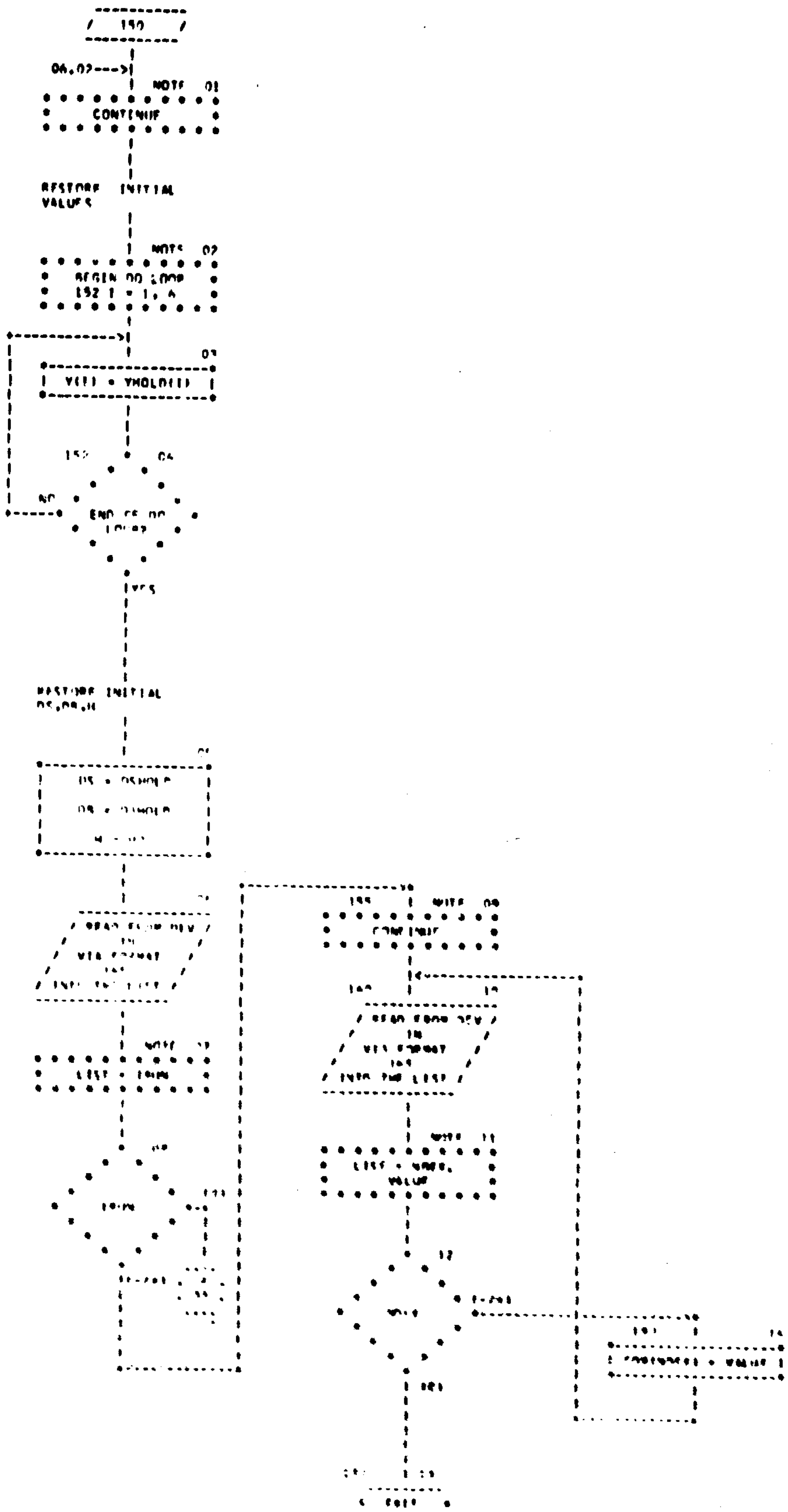
CHART TITLE - SUBROUTINE INPUT



03/11/69

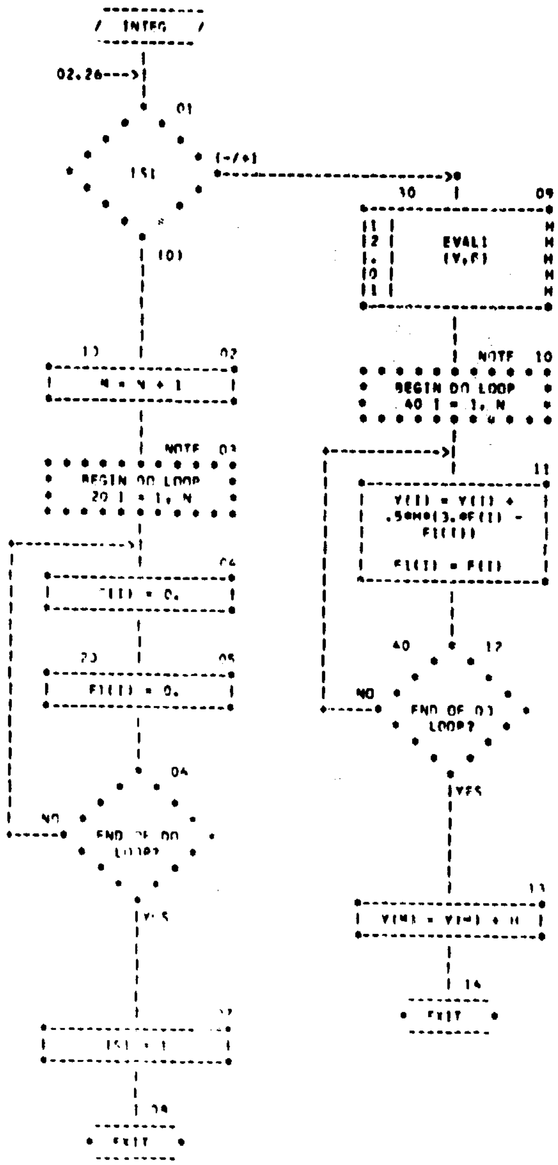
AUTOFLOW CHART SET - ... NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE INPUT



03/11/69 AUTOFLOW CHART SET - ZC790 NAVTRADEVCEN 68-C-0050-2

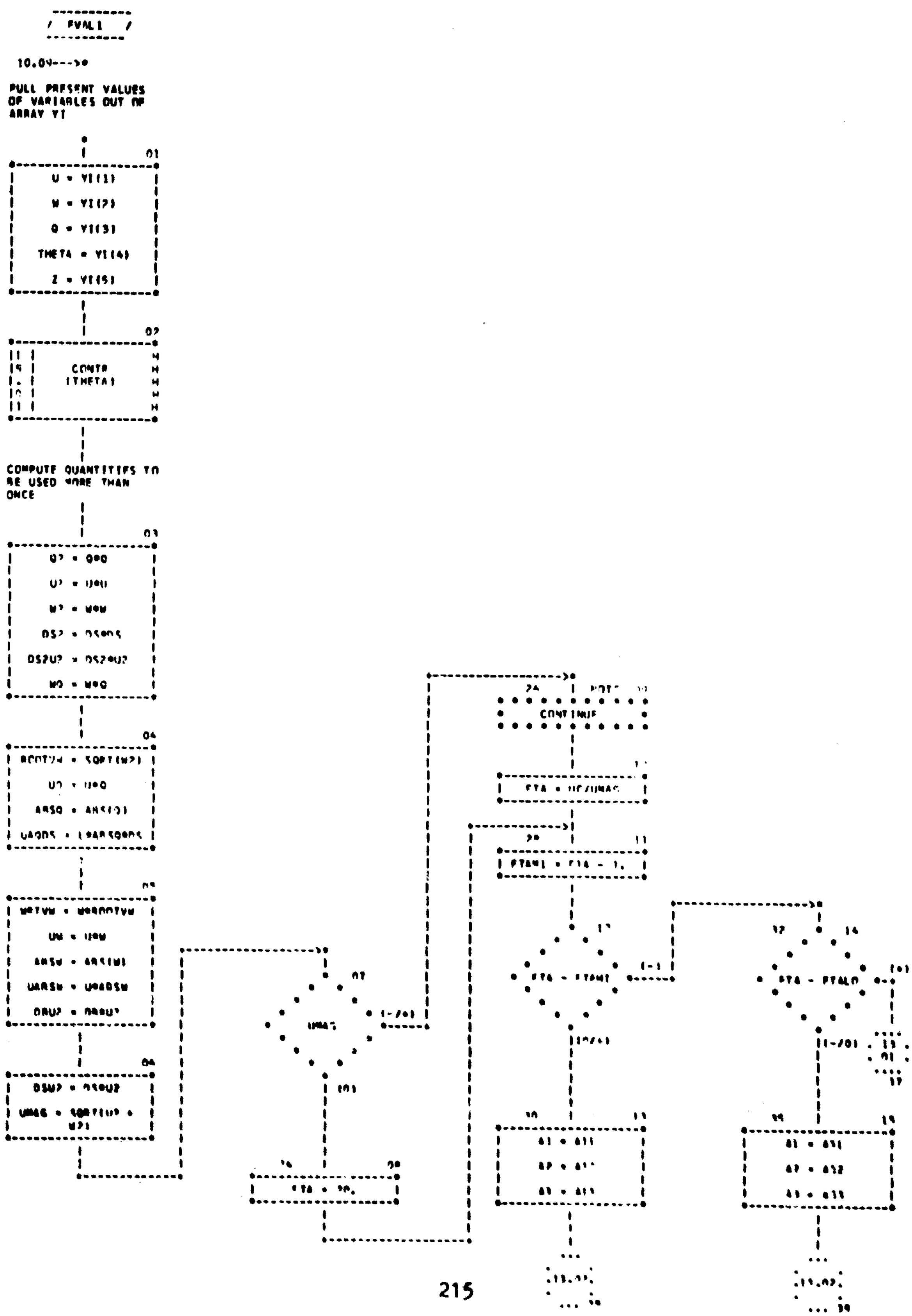
CHART TITLE - SUBROUTINE INTEG(Y)



03/11/69

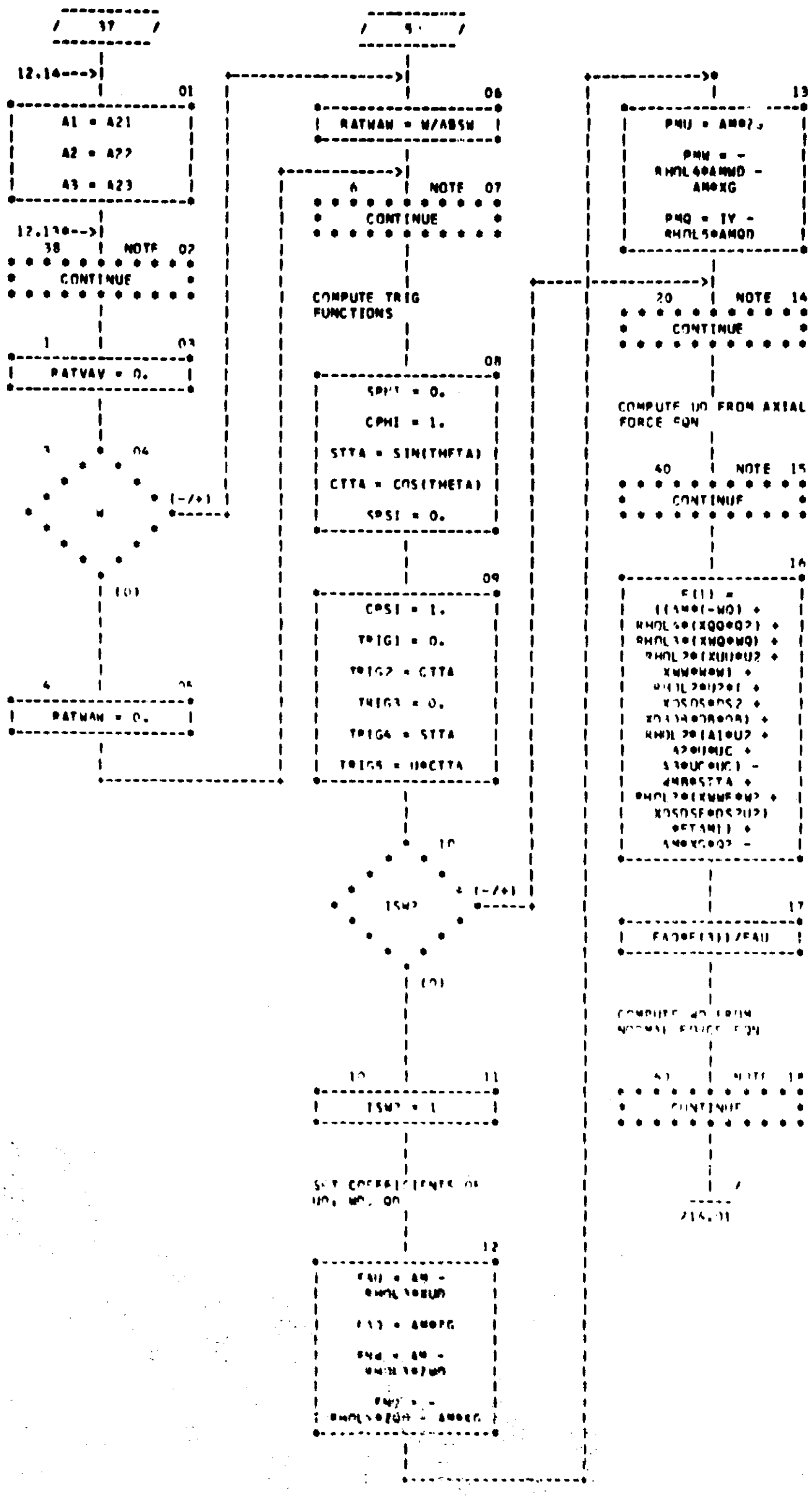
AUTOFLOW CHART SET - ZC790 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE FVAL1(YI,F)



03/11/69 AUTOFLOW CHART SET - 2C790 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE EVAL1(YI,F)



03/11/69

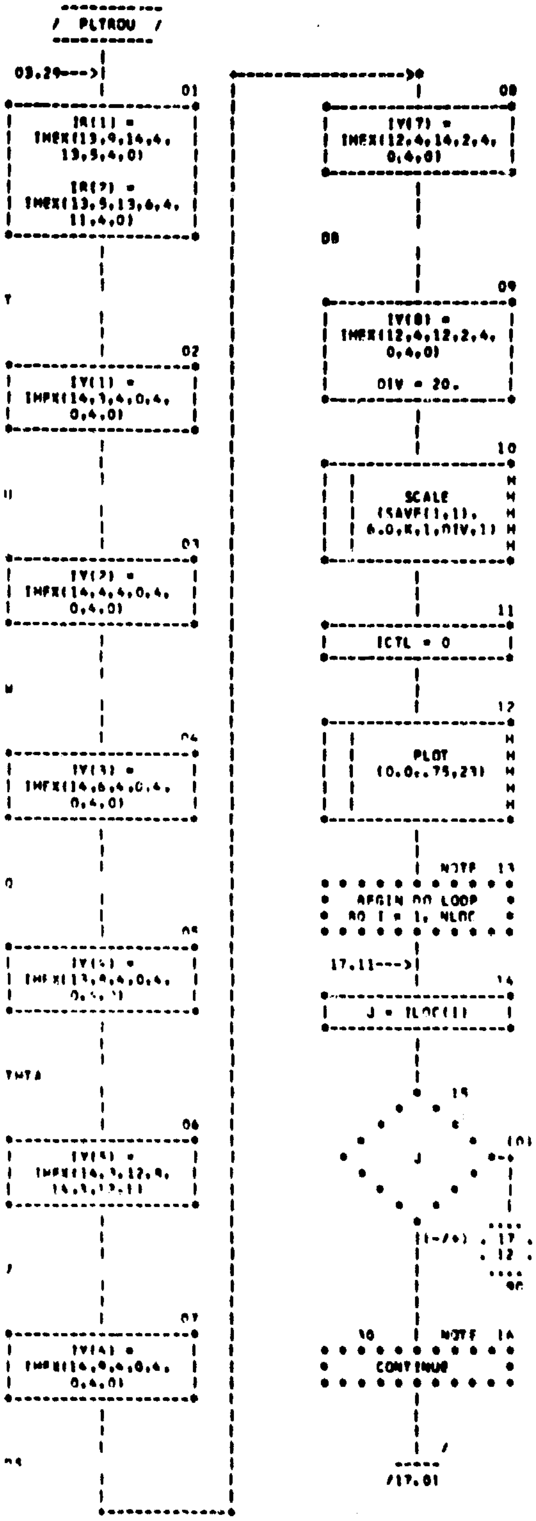
AUTOFLOW CHART SET - 2C790   NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE EVAL1(Y1,F)

```
13.1B--->*
         |                01
*------------------------*
|  F(2) = (AM*UQ +        |
|  RHOL3*(ZQ*UQ +         |
|   ZAQDS*UAQDS +         |
|   ZWAQ*RATWAW*          |
|  ROOTVW*ABSQ) +         |
| RHOL2*(ZSTR*U2 +        |
|     ZW*UW +             |
|   ZWAW*WRTVW +          |
|   ZAW*UABSW +           |
| ZWW*ABSW*ROOTVW +       |
|   ZDS*DSU2 +            |
|   ZDB*DBU2) +           |
|   WHR*TRIG2 +           |
|  RHOL3*ZQE*UQ*          |
|     ETAN1 +             |
|  RHOL2*(ZWE*UW +        |
|   ZWAWE*WRTVW +         |
*------------------------*
         |
         |                02
*------------------------*
|     ZDSE*DSU2)          |
|     *ETAN1 +            |
*------------------------*
         |
         |
COMPUTE QD FROM
PITCHING MOMENT EQN
         |
    PD   |   NOTE  03
* * * * * * * * * * * *
*     CONTINUE         *
* * * * * * * * * * * *
         |
         |                04
*------------------------*
|      F(3) =             |
| (RHOL5*(AM3AQ*Q*        |
|     ABSQ) +             |
|  RHOL4*(AMQ*UQ +        |
|   AMAQDS*UAQDS +        |
| AMAWQ*Q*ROOTVW) +       |
| RHOL3*(AMSTR*U2 +       |
|     AMW*UW +            |
|   AMWAW*WRTVW +         |
|   AMAW*UABSW +          |
| AMWW*ABSW*ROOTVW        |
|   + AMDS*DSU2 +         |
|   AMDB*DBU2) -          |
|     (XG*CW -            |
|  XB*CB)*TRIG2 -         |
|     (ZG*CW -            |
|  ZB*CB)*CSTTA +         |
*------------------------*
         |
         |                05
*------------------------*
|  RHOL4*AMQE*UQ*         |
|     ETAN1 +             |
*------------------------*
         |
         |
COMPUTE KINEMATICS -
THETA DOT
         |
         |                06
*------------------------*
|     F(6) = Q            |
*------------------------*
         |
         |
COMPUTE Z DOT
         |
         *----------------------------->*
                                        |                07
                              *------------------------*
                              | F(8) = - U*STTA +       |
                              |     W*TRIG2             |
                              *------------------------*
                                        |
                                        | 08
                                   ---------
                                   * EXIT  *
                                   ---------
```

03/11/69 AUTOFLOW CHART SET - IC790 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE PLTROU(SAVE,K,ILOC,NLOC,IRUN)



03/11/69 AUTOFLOW CHART SET - ZC790 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE PLTROU(SAVE,K,ILOC,NLOC,IRUN)

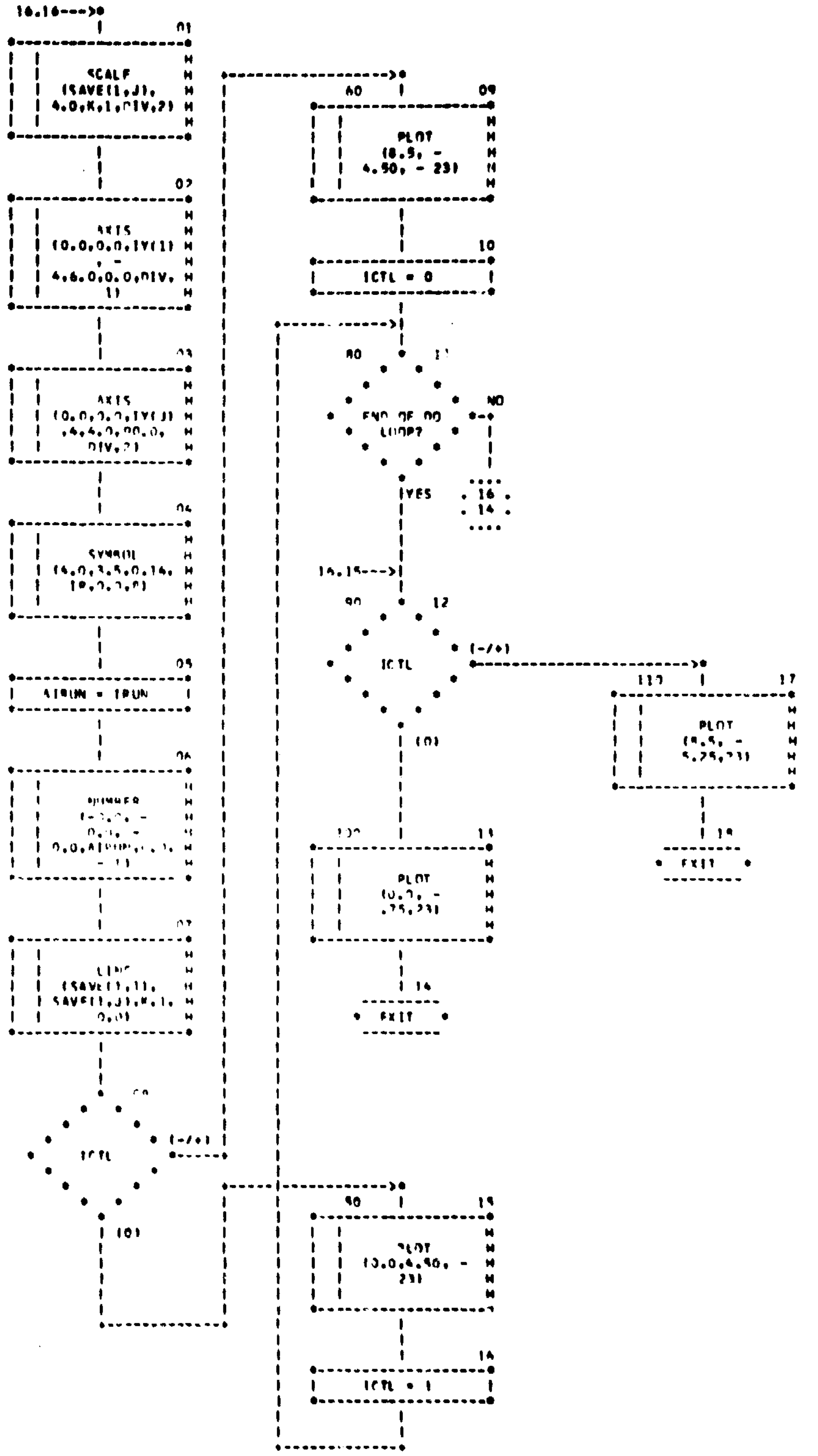
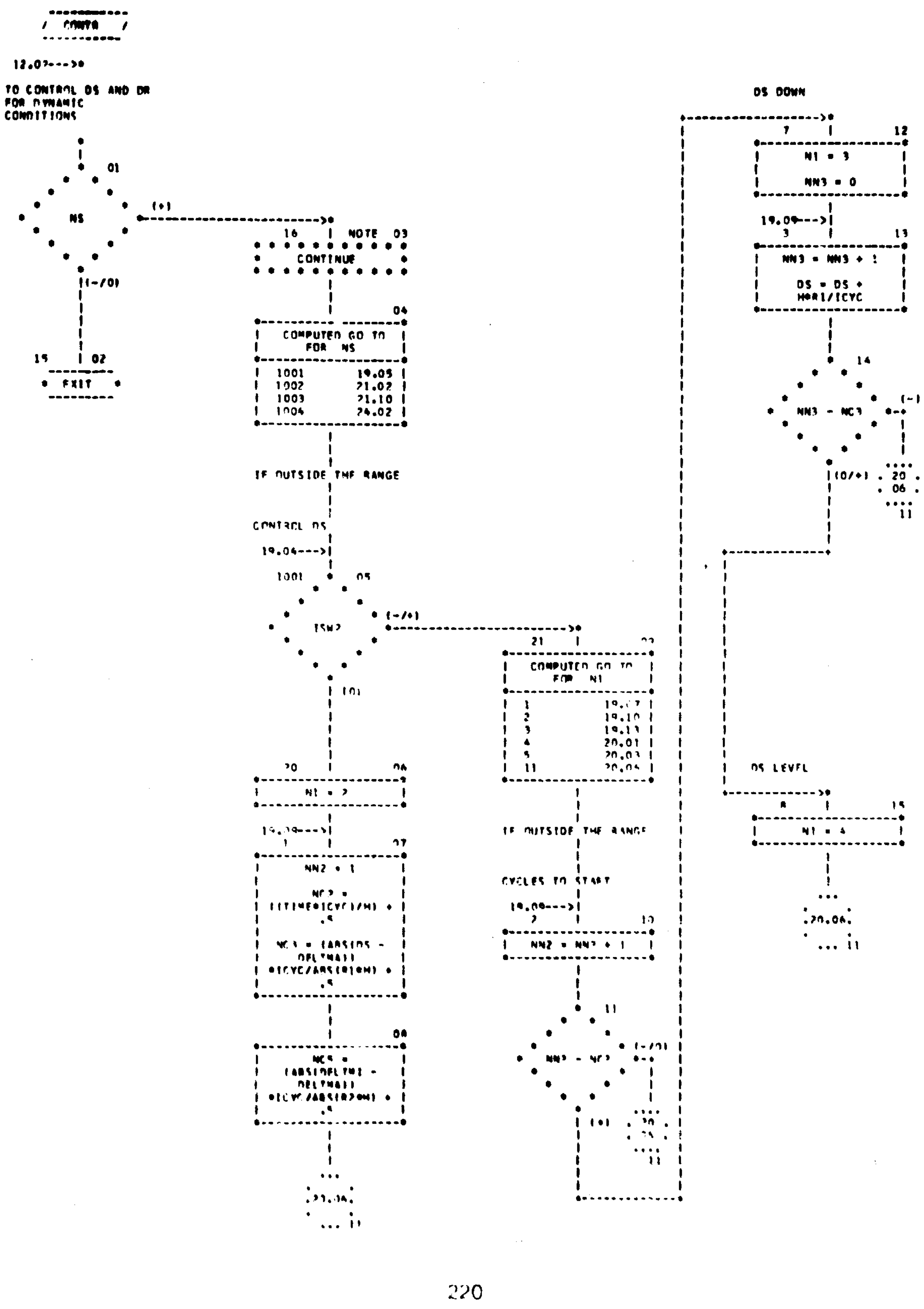
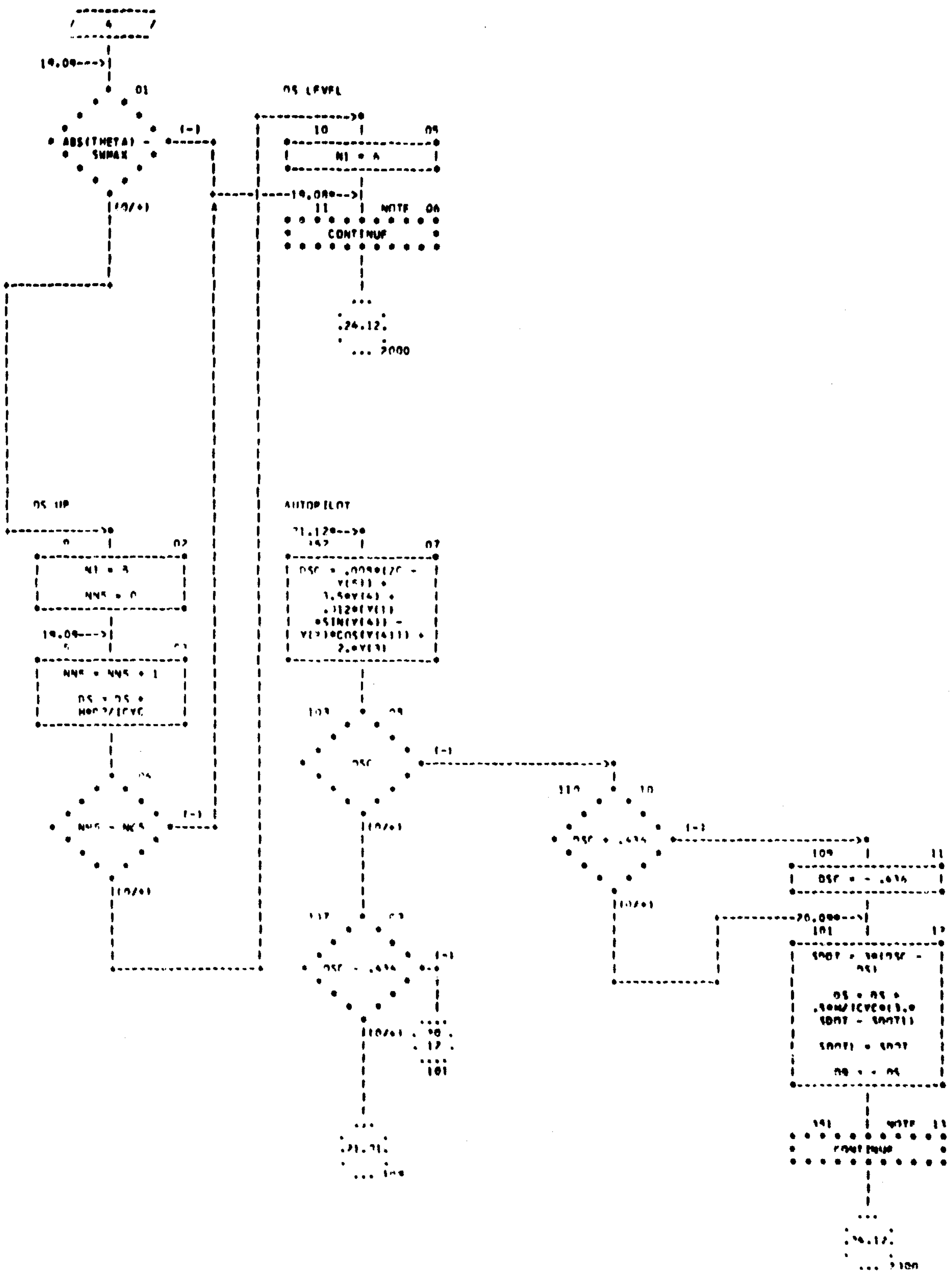


CHART TITLE - SUBROUTINE CONTR(THETA)



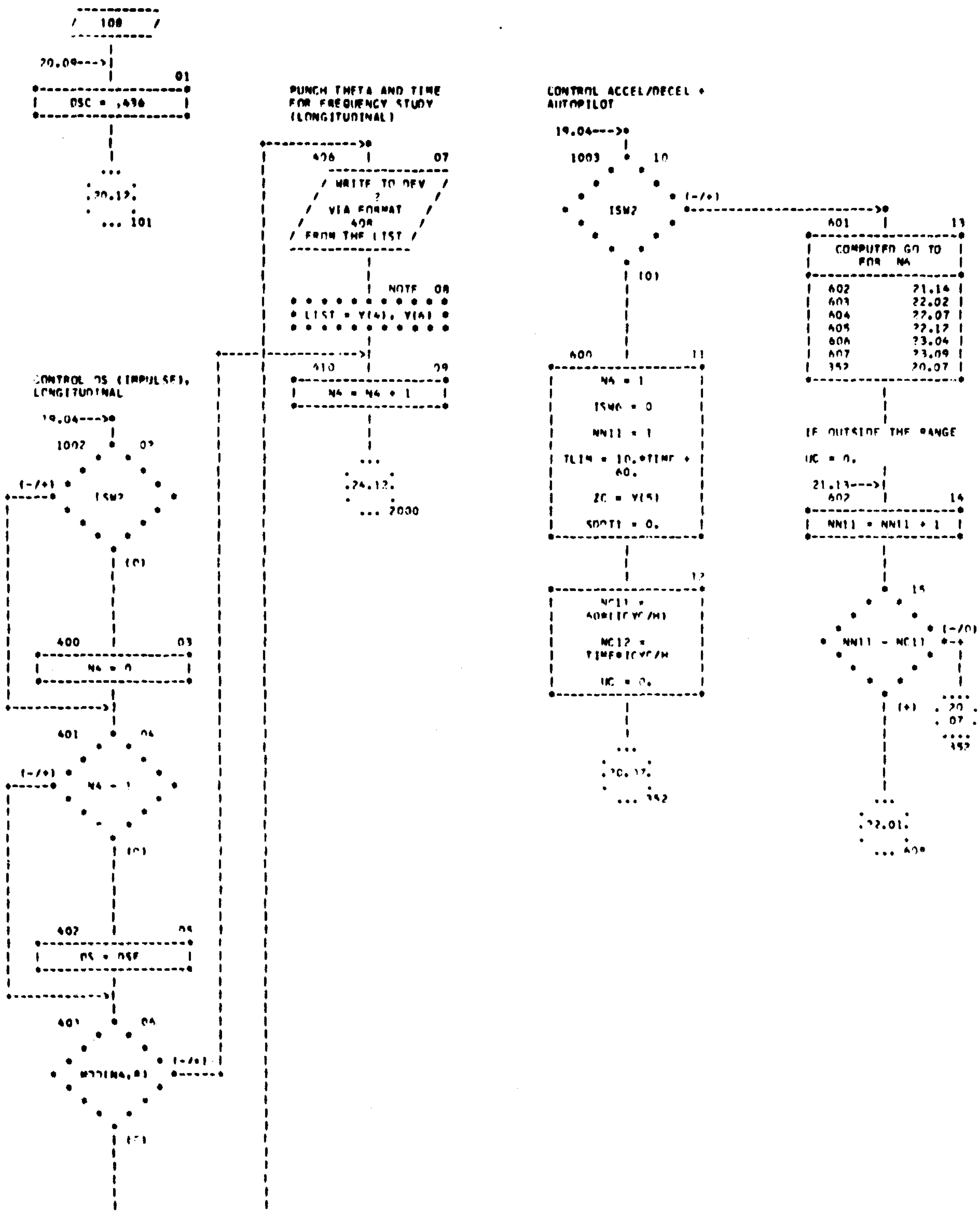
03/11/69 AUTOFLOW CHART SET - 2C790 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE CONTR(THETA)



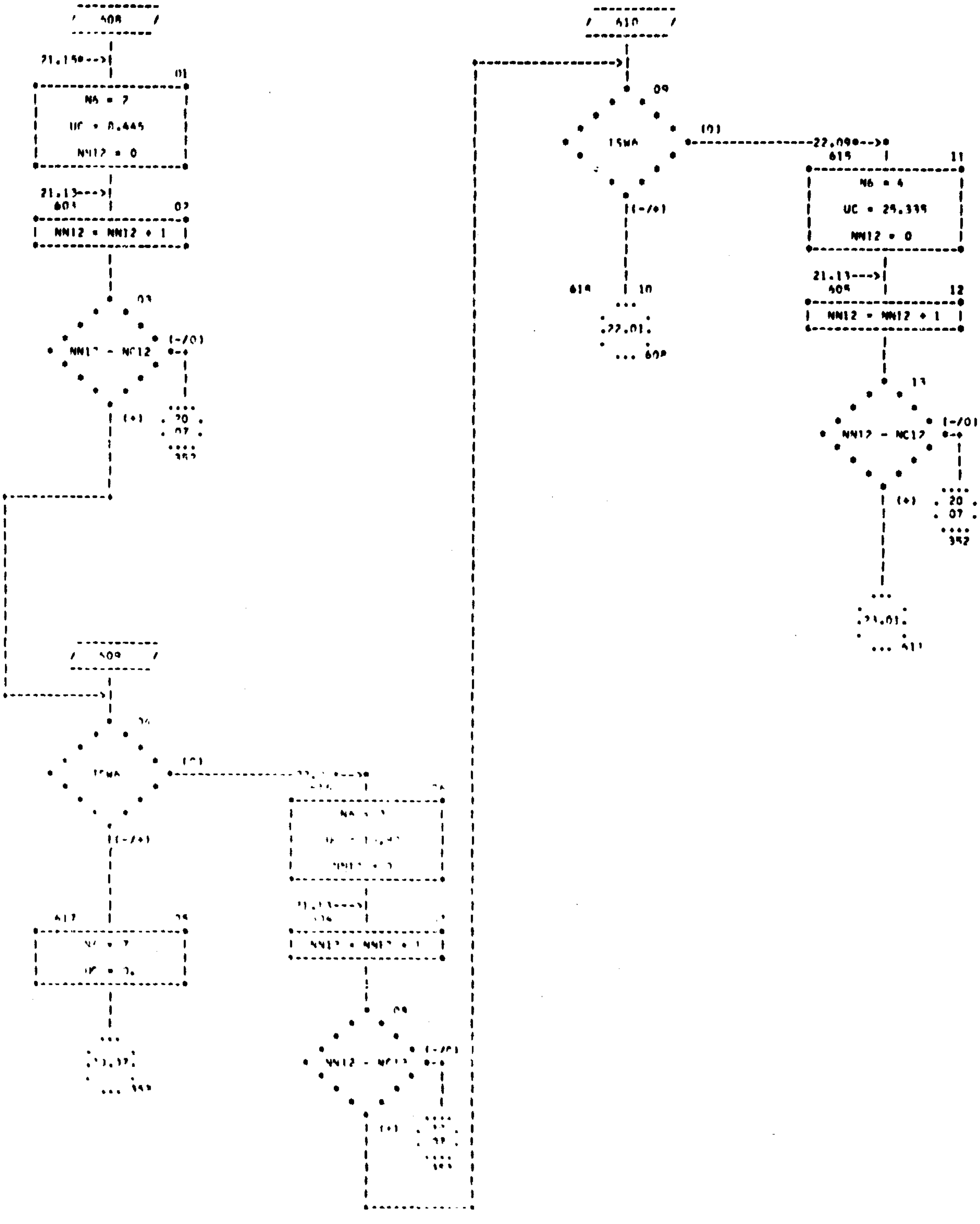
03/11/69 AUTOFLOW CHART SET - ZC790 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE CONTR(THETA)



03/11/69 AUTOFLOW CHART SET - 7C790 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE CONTRI(THETA)



03/11/69 AUTOFLOW CHART SET - 2C790 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE CONTR(THETA)

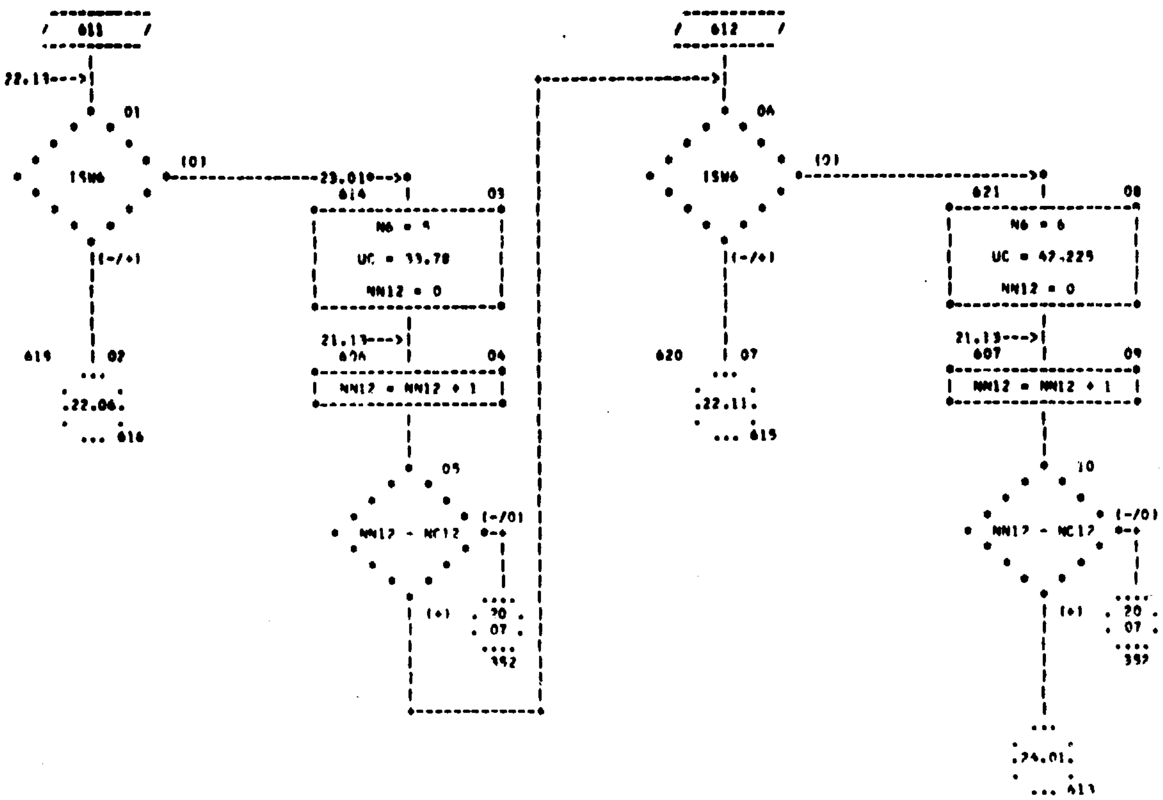
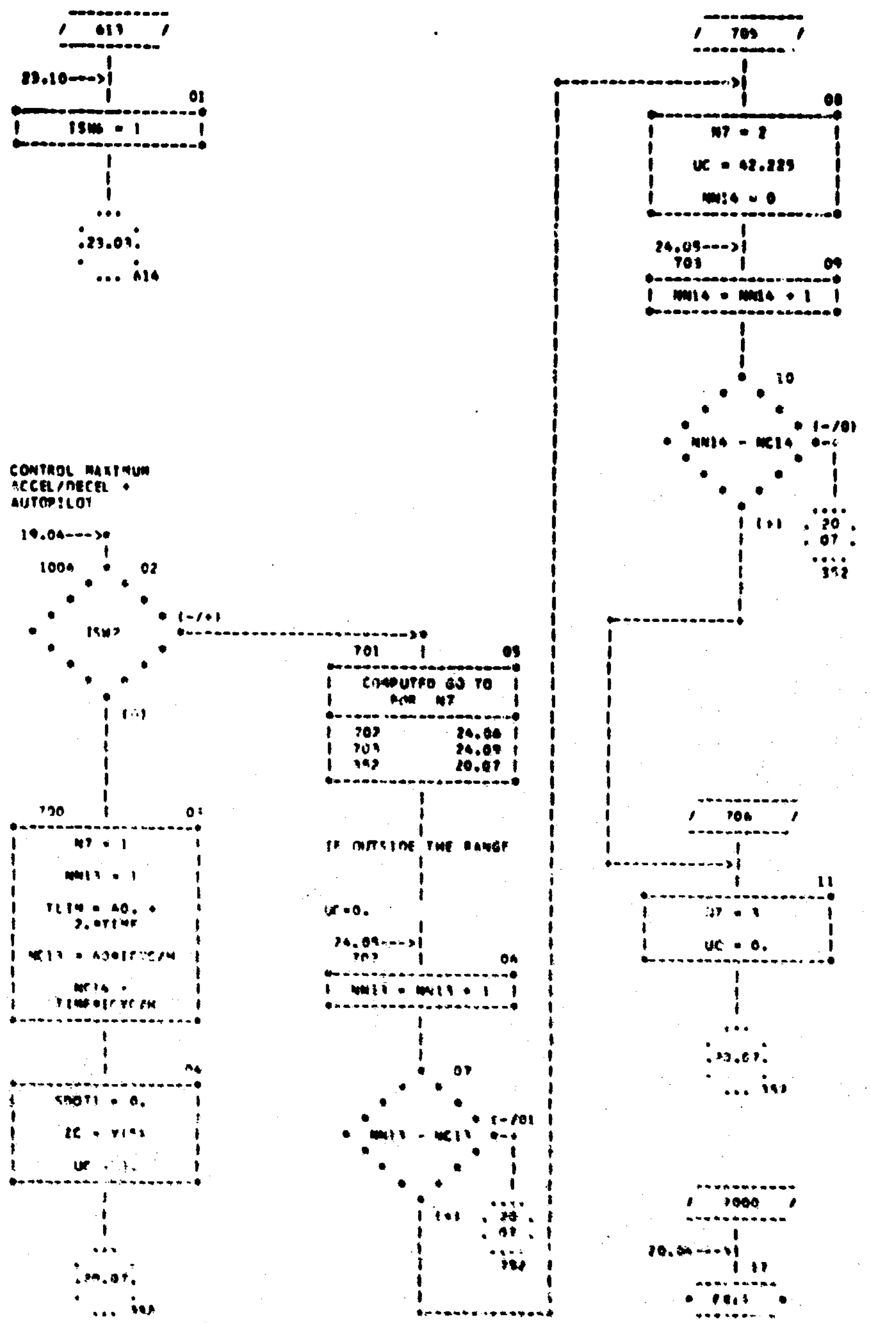


CHART TITLE - SUBROUTINE CONTR(THETA)

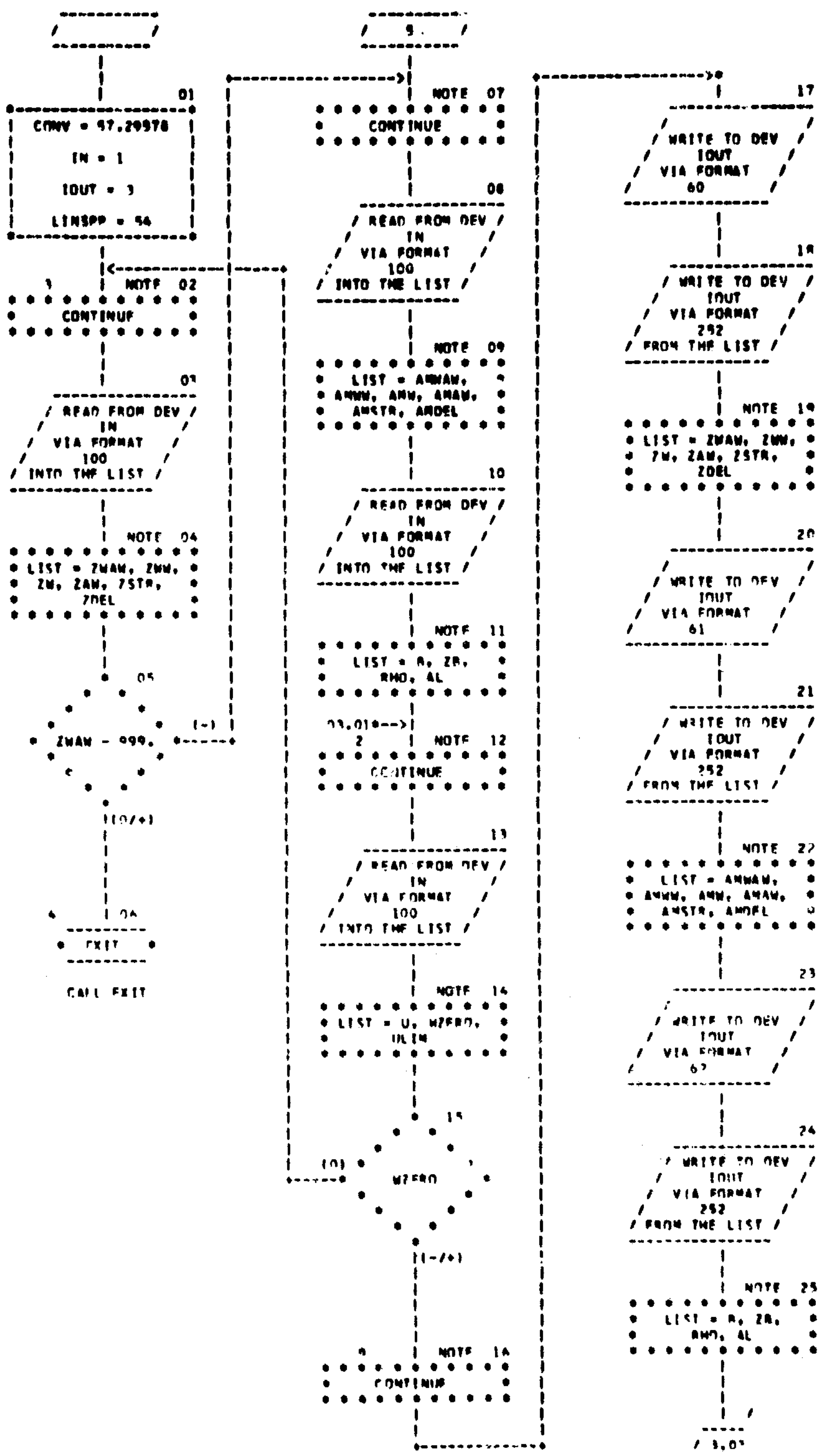


```
//        JOB   EC470
//        EXEC FFORTRAN
C
C   IN THE MANNER OF NEWTON
C
      CONV = 57.29578
      IN=1
      IOUT=3
      LINSPP=54
    3 CONTINUE
      READ(IN,100) ZWAW, ZWW, ZW, ZAW, ZSTR, ZDEL
      IF(ZWAW-999.)5,4,4
    4 CALL EXIT
    5 CONTINUE
      READ(IN,100) AMWAW,AMWW, AMW, AMAW, AMSTR, AMDEL
      READ(IN,100) B, ZB, RHO, AL
    2 CONTINUE
      READ(IN,100) U, WZERO ,ULIM
      IF(WZERO)9,3,9
    9 CONTINUE
      WRITE(IOUT,60)
      WRITE(IOUT,252)ZWAW,ZWW,ZW,ZAW,ZSTR,ZDEL
      WRITE(IOUT,61)
      WRITE(IOUT,252)AMWAW,AMWW,AMW,AMAW,AMSTR,AMDEL
      WRITE(IOUT,62)
      WRITE(IOUT,252) B, ZB, RHO, AL
   60 FORMAT(1H1,50X,'EC470 - INITIAL CONDITIONS'//
     11H ,3X,'ZWAW',14X,'ZWW',11X,'ZW',14X,'ZAW',12X,'ZSTR',13X,
     2'ZDEL')
   61 FORMAT(1H ,3X,'MWAW',14X,'MWW',11X,'MW',14X,'MAW',12X,'MSTR',13X,
     2'MDEL')
   62 FORMAT(1H ,3X,'B',17X,'ZB',12X,'RHO',13X,'L')
      WRITE(IOUT,215)
      LINS = 9
  100 FORMAT(8F10.5)
    8 CONTINUE
      ICNT = 0
      WWK=WZERO
      U2=U*U
      R = ZDEL/AMDEL
      C1=  ZWAW-R*AMWAW
      C2=  ZWW-R*AMWW
      C3= (ZW -R*AMW)*U
      C4= (ZAW-R*AMAW)*U
      C5= (ZSTR-R*AMSTR)*U2
      COEF=2.*B*ZB/(RHO*AL*AL*AL)
      C6 = -R*COEF
   10 CONTINUE
      ICNT = ICNT + 1
      IF(ICNT - 200) 15,15,12
   12 WRITE(IOUT,13)
   13 FORMAT(1H , 'ITERATIONS EXCEED 200')
      GO TO 2
   15 CONTINUE
```

```
      ABSW=ABS(WWK)
      W2=WWK*WWK
      WABSW=WWK*ABSW
      ABSWU=ABSW*U
      WU=WWK*U
      T1 = W2+U2
      ROOT = SQRT(T1)
      T2=COEF*WWK/ROOT
      F = C1*WABSW+C2*W2+C3*WWK+C4*ABSW +C5+C6*WWK/ROOT
      IF(WWK)20,18,30
   18 WRITE(IOUT,19)
   19 FORMAT(1H ,'W IS ZERO')
      GO TO 2
   20 S=-1.
      GO TO 40
   30 S= 1.
   40 CONTINUE
      FP =2.*S*C1*WWK +2*C2*WWK+C3+S*C4+C6*U2/(ROOT*T1)
      WS = WWK- F/FP
      IF(ABS(WWK-WS)-.0001*ABS(WWK))200,200,150
  150 WWK = WS
      GO TO  10
  200 CONTINUE
      DS=-(AMWAW*WABSW+AMWW*W2+AMW*WU+AMAW*ABSWU+
     1 AMSTR*U2+T2)/(AMDEL*U2)
      IF(LINSPP-LINS)210,210,250
  210 WRITE(IOUT,211)
  211 FORMAT(1H1)
      WRITE(IOUT,215)
  215 FORMAT(/1H ,3X,'U(FT/SEC)',9X,'U(KTS)',8X,'W(FT/SEC)',7X,
     1 'DEL(RAD)',7X,'THETA(RAD)',7X,'DEL(DEG)',7X,'THETA(DEG)'/)
      LINS=0
  250 LINS=LINS+1
      UKTS=U/1.689
      THTA = ATAN(WS/U)
      DS1=DS*CONV
      THTAD=THTA*CONV
      WRITE(IOUT,252)U,UKTS,WS,DS ,THTA,DS1,THTAD
  252 FORMAT(1H ,F13.6,6F16.6)
      WZERO = WS
      U = U - 1.689
      IF(U-ULIM) 2,8,8
  260 GO TO 2
      END
/*
/&
```

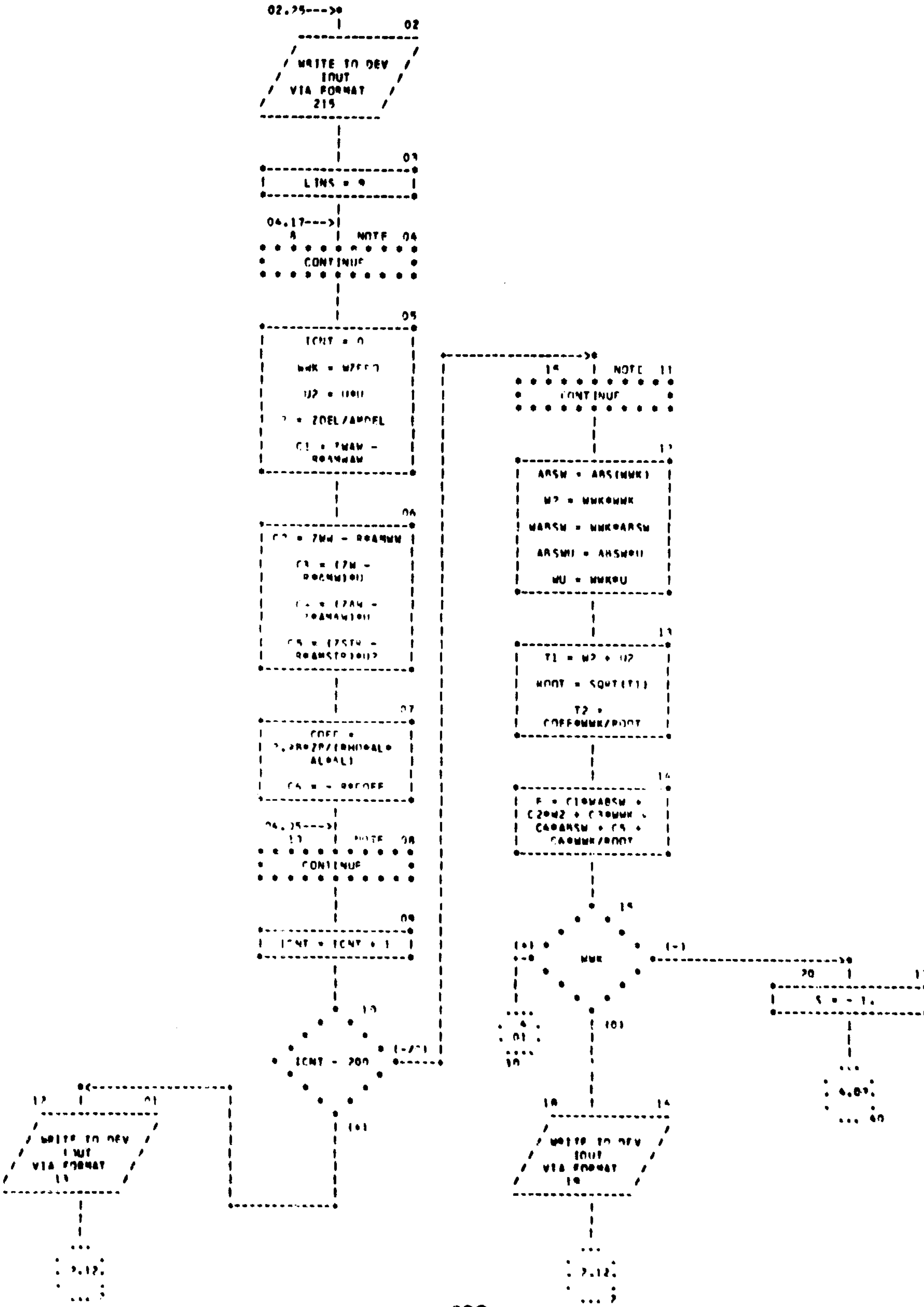
03/11/69 AUTOFLOW CHART SET - EC470 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES



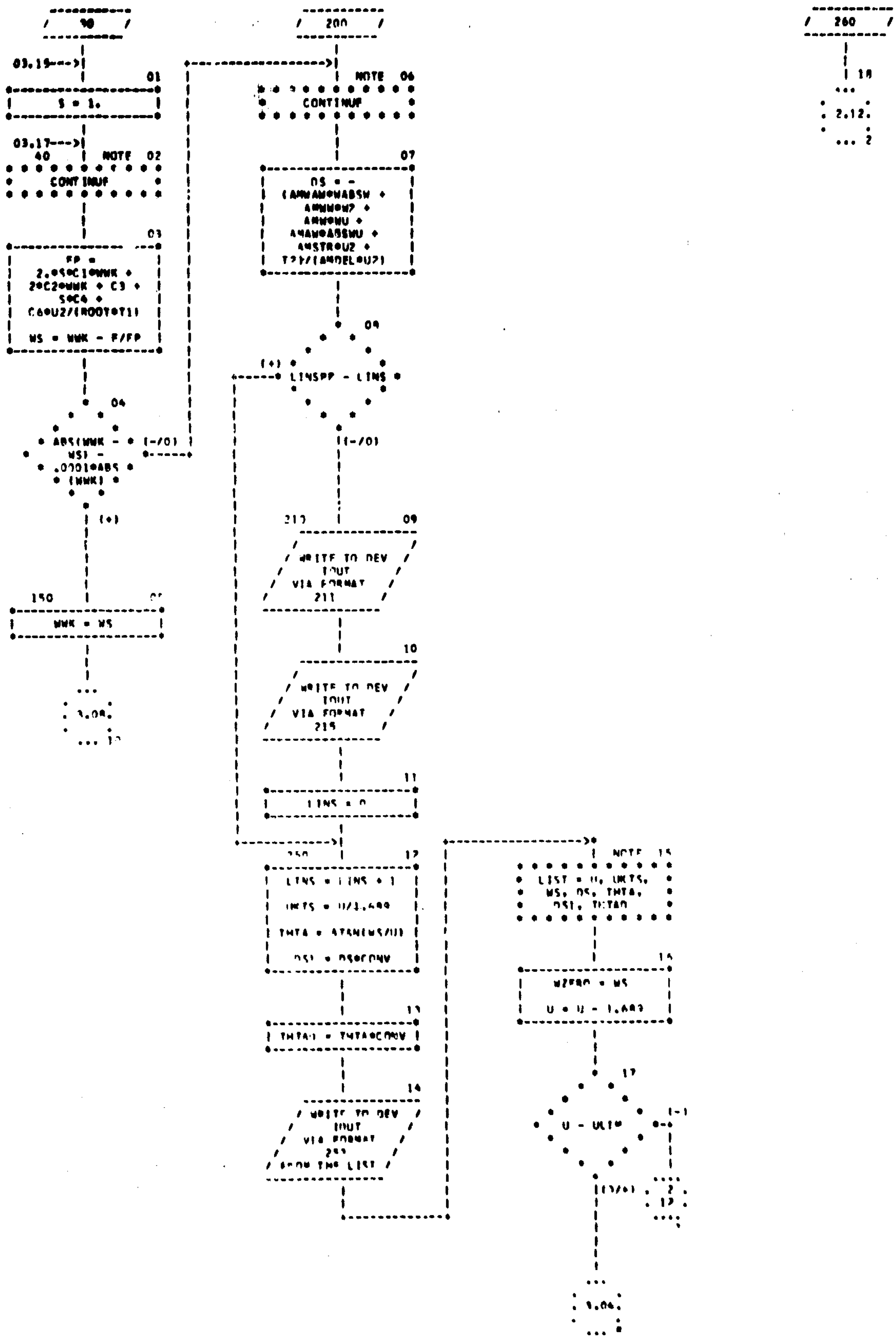
03/11/69 AUTOFLOW CHART SET - FC470 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES



03/11/69 AUTOFLOW CHART SET - EC470 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES



```
//       JOB   EC430
//       EXEC FFORTRAN
      DIMENSION    ICTL(50),  W(50), X(50), Y(50), Z(50), INTS(50)
      IN = 1
      IOUT = 3
      LINSPP = 54
      LINS = 99
      IPAGE = 1
      READ(IN,95) N, IPNT
      DO 90 I = 1, N
      INTS(I) = I
   90 CONTINUE
   95 FORMAT(16I5)
      READ(IN,100) (W(I), X(I), Y(I), Z(I), I = 1,N)
      IF(IPNT) 96, 99,96
   96 WRITE(IOUT,190) IPAGE
      IPAGE = IPAGE+1
      WRITE(IOUT,97)
   97 FORMAT(1H ,8X,'N',10X,'W',15X,'X',15X,'Y',15X,'Z'/)
      WRITE(IOUT,98)  (INTS(I),W(I),X(I),Y(I),Z(I),I=1,N)
   98 FORMAT(1H ,I9,1X,4E16.6)
   99 CONTINUE
  100 FORMAT(8F10.5)
  105 READ(IN,110)(ICTL(I),I=1,N)
  110 FORMAT(80I1)
      IF(ICTL(1)-2)130,120,120
  120 CALL EXIT
  130 CONTINUE
      SW = 0.
      SWX = 0.
      SWY = 0.
      SWZ = 0.
      DO 170 I = 1,N
      IF(ICTL(I))160,170,160
  160 TW = W(I)
      SW = SW + TW
      SWX = SWX + TW * X(I)
      SWY = SWY + TW * Y(I)
      SWZ = SWZ + TW * Z(I)
  170 CONTINUE
      XG = SWX/SW
      YG = SWY/SW
      ZG = SWZ/SW
      IF(LINSPP-LINS)180,180,200
  180 WRITE(IOUT,190) IPAGE
  190 FORMAT(1H1,3X,'EC430',40X,'COMPUTED  CENTER  OF  GRAVITY',
     1 30X,'PAGE',I5/)
      LINS = 2
      IPAGE = IPAGE + 1
  200 WRITE(IOUT,210)(INTS(I),I=1,N)
  210 FORMAT(1H ,9I3,24I4)
      WRITE(IOUT,210)(ICTL(I),I=1,N)
      WRITE(IOUT,220) XG, YG, ZG, SW
  220 FORMAT(/1H ,8X,'XG',14X,'YG',14X,'ZG',14X,'W'/
```

```
      11H ,4E16.6/)
       LINS = LINS + 6
       GO TO 105
       END
/*
/&
```

03/11/69 AUTOFLOW CHART SET - EC430 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES

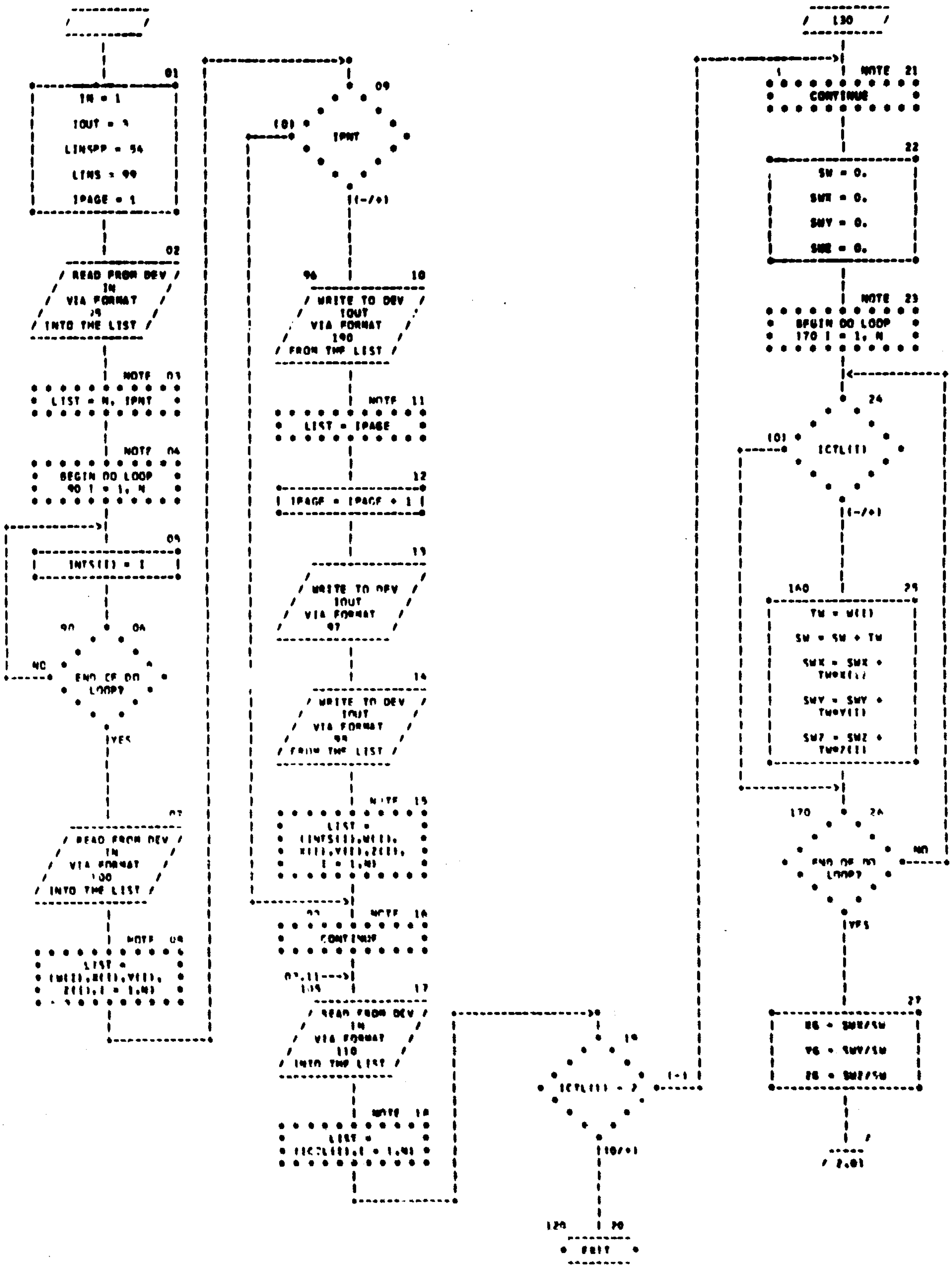
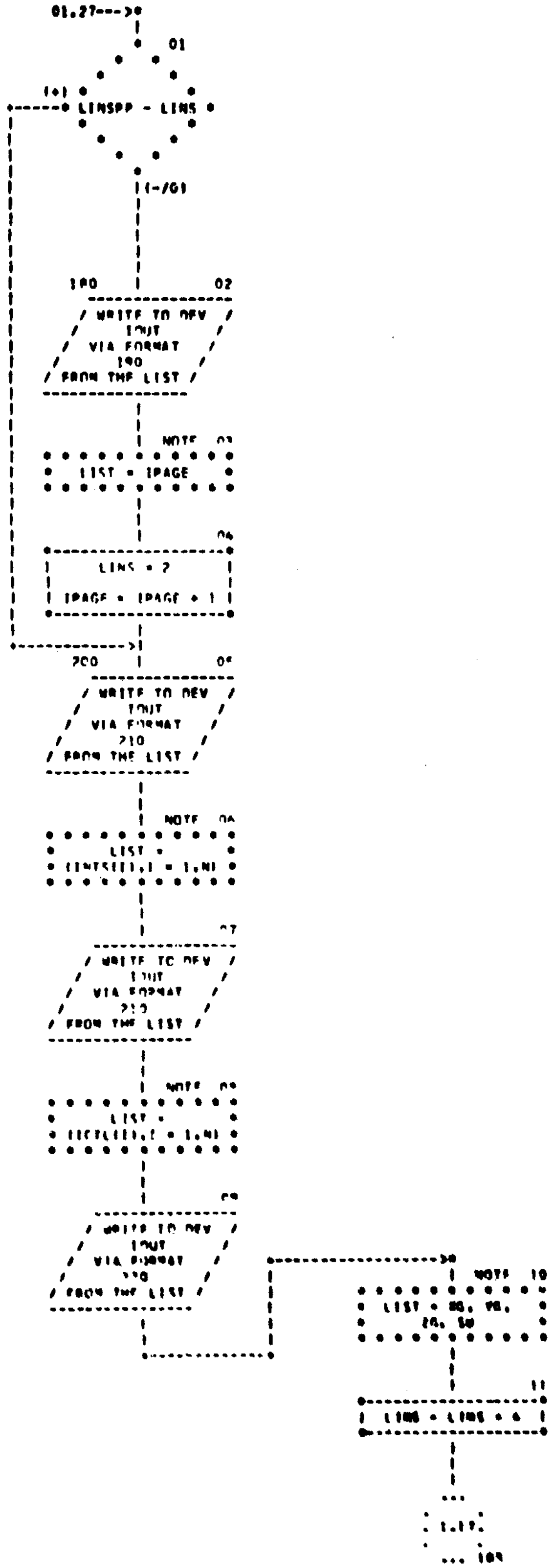


CHART TITLE - PROCEDURES



```
//        JOB    ZC300
//        EXEC FFORTRAN
C      SUBMARINE THRUST
       DIMENSION THRUST(10)
       READ(1,30)AL,AM,ETAHI,ETALO
       READ(1,40)A1,A2,A3
       READ(1,40)B1,B2,B3
       READ(1,40)C1,C2,C3
    30 FORMAT(8F10.3)
    40 FORMAT(8F10.6)
       WRITE(3,36)
    36 FORMAT(1H1,5X,23HZC300, SUBMARINE THRUST//T6,'L',T16,'M',T24,'ETAH
      2I',T34,'ETALO')
       WRITE(3,41)AL,AM,ETAHI,ETALO
    41 FORMAT(F9.2,F11.2,2F10.3)
       WRITE(3,42)
    42 FORMAT(1HO,T10,'A1',T20,'A2',T30,'A3')
       WRITE(3,43)A1,A2,A3
    43 FORMAT(5X,3F10.6)
       WRITE(3,44)
    44 FORMAT(1HO,T10,'B1',T20,'B2',T30,'B3')
       WRITE(3,43)B1,B2,B3
       WRITE(3,45)
    45 FORMAT(1HO,T10,'C1',T20,'C2',T30,'C3')
       WRITE(3,43)C1,C2,C3
       J=1
       DO 25 K=1,2
       WRITE(3,26)
    26 FORMAT(1H1)
       IF(J)32,33,33
    32 WRITE(3,34)
    34 FORMAT(38X,40HACCELERATION - (FEET PER SECOND-SQUARED)/)
       GO TO 35
    33 WRITE(3,31)
    31 FORMAT(50X,16HFORCE - (POUNDS)/)
    35 UC1=-20.
       DO 8 N=1,10
       UC1=UC1+5.
     8 THRUST(N)=UC1
       WRITE(3,12)THRUST
    12 FORMAT(5X,2HUC,5X,10F11.2)
       WRITE(3,11)
    11 FORMAT(1HO,6X,1HU)
       U1=-2.5
       DO 7 M=1,13
       U1=U1+2.5
    20 U=U1*1.689
       UC1=-20.
       DO 24 N=1,10
       UC1=UC1+5.
       UC=UC1*1.689
       IF(U)51,52,51
    51 ETA=UC/U
       GO TO 55
```

```
 52 IF(UC)54,53,53
 53 ETA=1.
    GO TO 55
 54 ETA=-1.
 55 RHOL=.9975*AL*AL
    IF(ETA-ETAHI)2,3,3
  2 IF(ETA-ETALO)4,5,5
  3 TX=RHOL*(A1*U*U+B1*U*UC+C1*UC*UC)
    GO TO 6
  4 TX=RHOL*(A3*U*U+B3*U*UC+C3*UC*UC)
    GO TO 6
  5 TX=RHOL*(A2*U*U+B2*U*UC+C2*UC*UC)
  6 IF(J)23,23,24
 23 TX=TX/AM
 24 THRUST(N)=TX
    IF(J)39,39,37
 39 WRITE(3,10)U1,THRUST
 10 FORMAT(1H0,F11.2,10F11.4)
    GO TO 7
 37 WRITE(3,38)U1,THRUST
 38 FORMAT(1H0,11F11.2)
  7 CONTINUE
 25 J=-1
    END
/*
/&
```

03/11/69 AUTOFLOW CHART SET - ZC300 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES

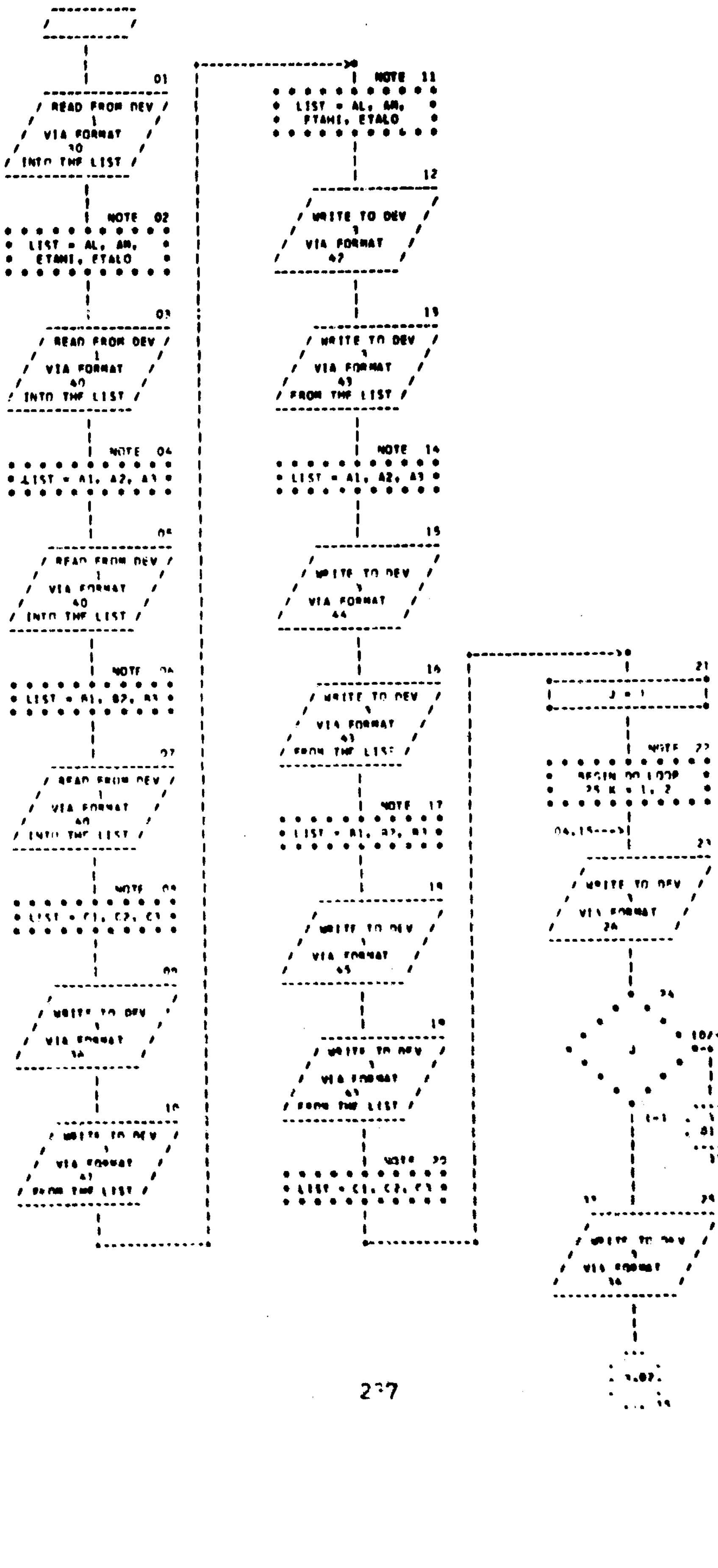


CHART TITLE - PROCEDURES

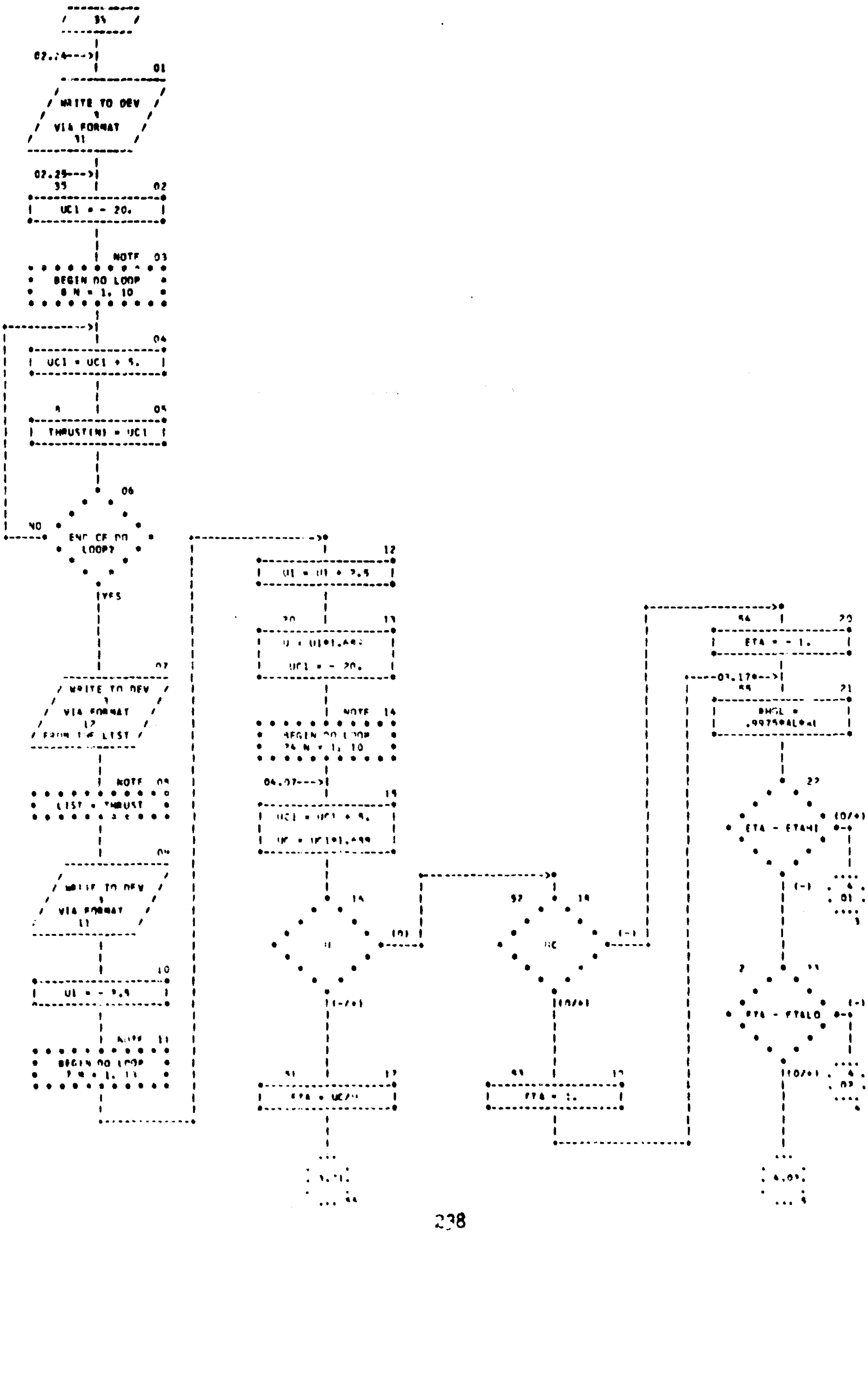
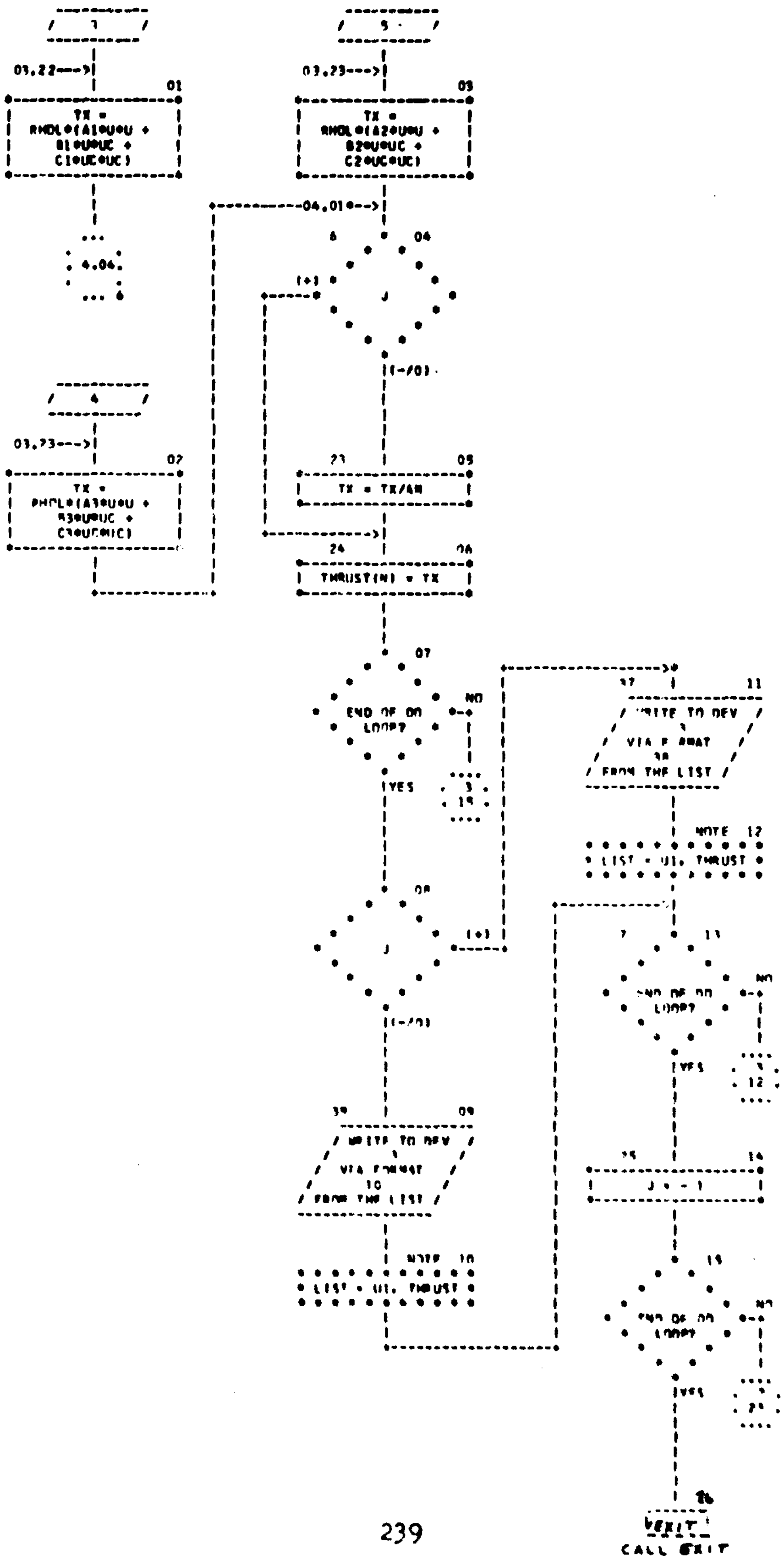


CHART TITLE - PROCEDURES



```
//        JOB   ZC690
//        EXEC FFORTRAN
C
C  ZC690  ERROR CALCULATOR, DS + DR CONTROL
C
      DIMENSION COEF(2)
      WRITE(3,14)
   14 FORMAT(1H1,T5,'ZC690, ERROR CALCULATIONS, DS + DR CONTROL')
      NS = 1
      ISW=1
      WRITE(3,11)
   11 FORMAT(1H0,T3,'NO.',T7,'SPEED',T14,'COEF',T29,'PERCENT CHANGE OF V
     1ARIABLE',T93,'INPUT DATA', / T7,'(KTS)',T23,'DU',T30,'DTHETA',T38,
     2'DPHI1',T46,'DPHI2',T54,'DPSI',T73,'U60',T83,'THETHI',T95,'PHIMIN'
     3,T107,'PHIUP',T119,'PSI60'//)
      READ(1,15)NO15,NO25
   15 FORMAT(I4,6X,I4)
    1 READ(1,10)NO,COEF,U60,THETHI,PHIMIN,PHIUP,PSI60
   10 FORMAT(I4,6X,2A4,2X,F10.5,F10.6,2F10.8,F10.6)
      IF(NO-1)100,6,6
    6 GO TO (2,3,4),NS
    2 IF(NO-NO15)21,22,22
   21 IF(ISW)4,4,23
   23 ISPEED=5
      PHIO=0.
      PSIO=0.
      U0=8.445
      THETO=.005155
    7 DUR=U0-U60
      DTHETR=THETO-THETHI
      DPHI1R=PHIO-PHIMIN
      DPHI2R=PHIO-PHIUP
      DPSIR=PSIO-PSI60
      ISW=0
      WRITE(3,12)NO,ISPEED,COEF,U60,THETHI,PHIMIN,PHIUP,PSI60
   12 FORMAT(1H0,T2,I4,T9,I2,T14,2A4,T70,F9.5,T82,F8.6,T93,F10.8,T105,
     1F10.8,T117,F9.6)
      GO TO 1
    4 DU=U0-U60
      DTHET=THETO-THETHI
      DPHI1=PHIO-PHIMIN
      DPHI2=PHIO-PHIUP
      DPSI=PSIO-PSI60
      DELU=100.*(DUR-DU)/DUR
      DELTH=100.*(DTHETR-DTHET)/DTHETR
      DELPH1=100.*(DPHI1R-DPHI1)/DPHI1R
      DELPH2=100.*(DPHI2R-DPHI2)/DPHI2R
      DELPSI=100.*(DPSIR-DPSI)/DPSIR
      WRITE(3,13)NO,ISPEED,COEF,DELU,DELTH,DELPH1,DELPH2,DELPSI,U60,
     1THETHI,PHIMIN,PHIUP,PSI60
   13 FORMAT(T2,I4,T9,I2,T14,2A4,T22,F5.1,T30,F5.1,T38,F5.1,T46,F5.1,T54
     1,F5.1,T70,F9.5,T82,F8.6,T93,F10.8,T105,F10.8,T117,F9.6)
      GO TO 1
   22 ISW=1
```

```
      NS=2
    3 IF(NO-NO25)24,25,25
   24 IF(ISW)4,4,26
   26 ISPEED=15
      U0=25.335
      THETO=.001895
      GO TO 7
   25 ISW=1
      NS=3
      IF(ISW)4,4,27
   27 ISPEED=25
      U0=42.225
      THETO=.001803
      GO TO 7
  100 CALL EXIT
      END
/*
/&
```

03/11/69

AUTOFLOW CHART SET - ZC600 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES

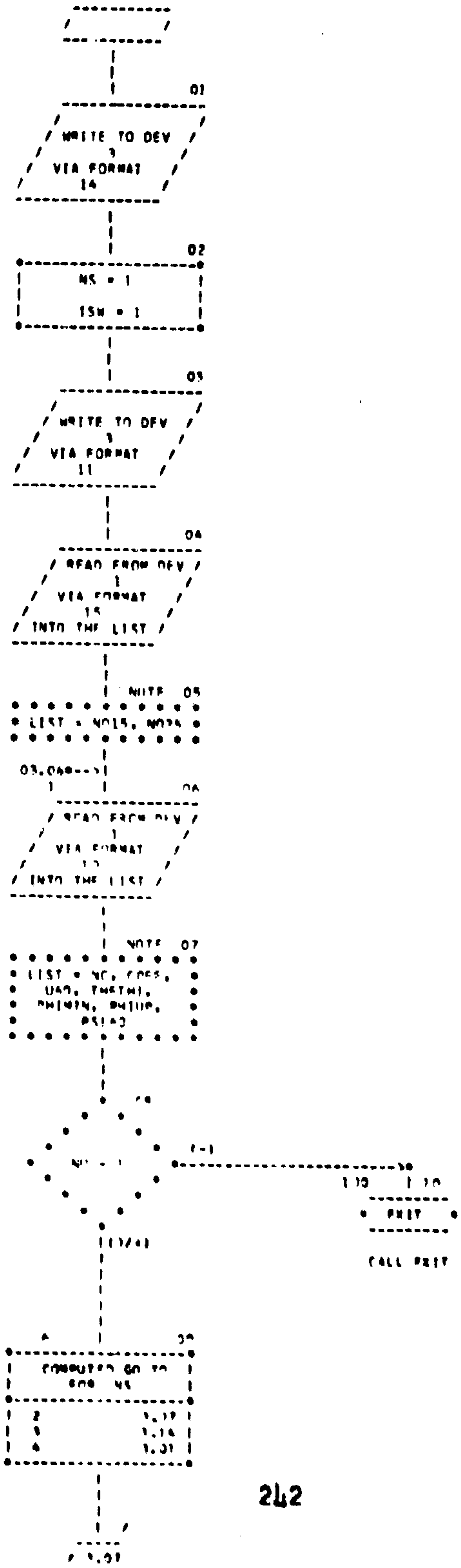
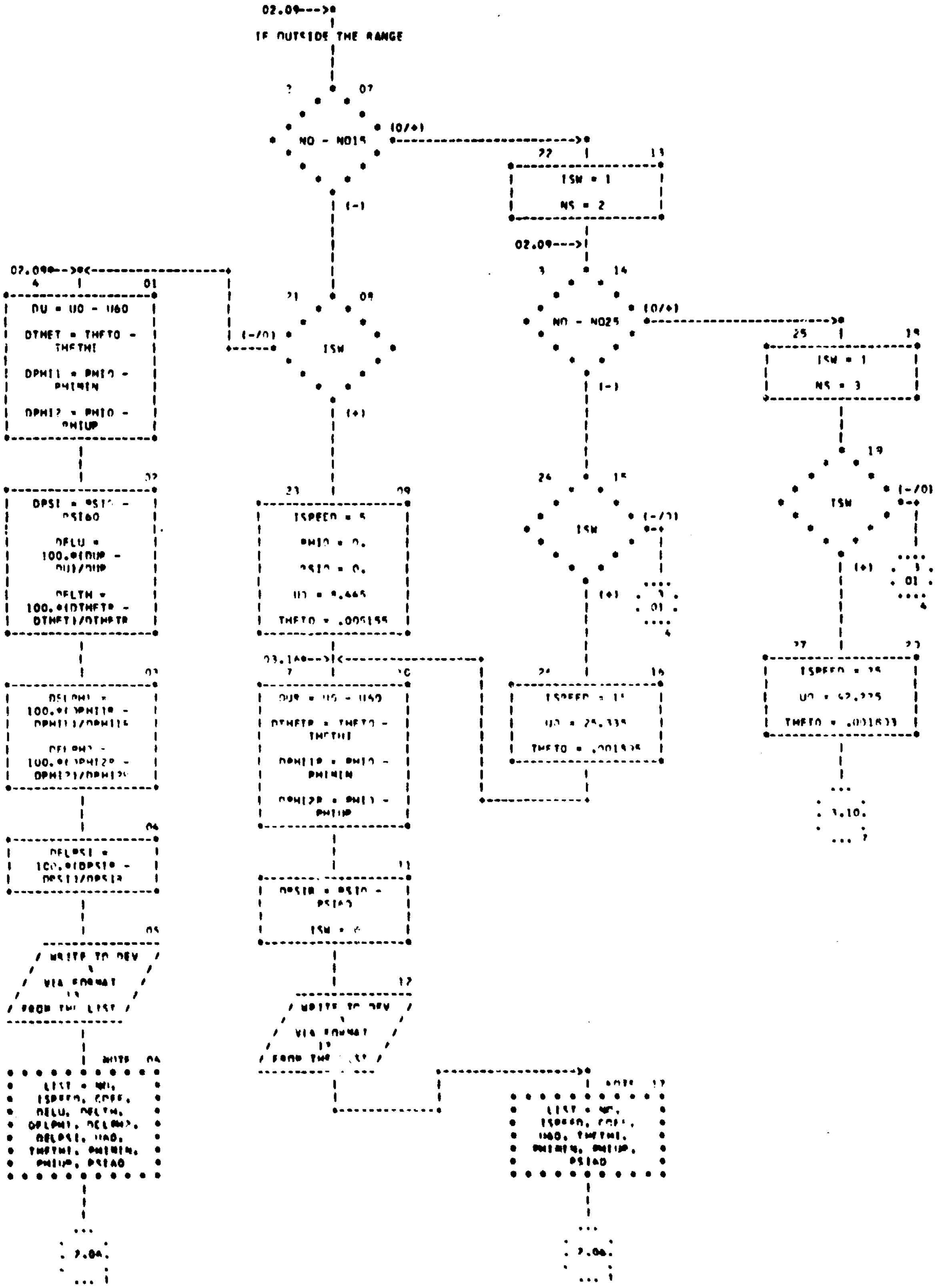


CHART TITLE - PROCEDURES

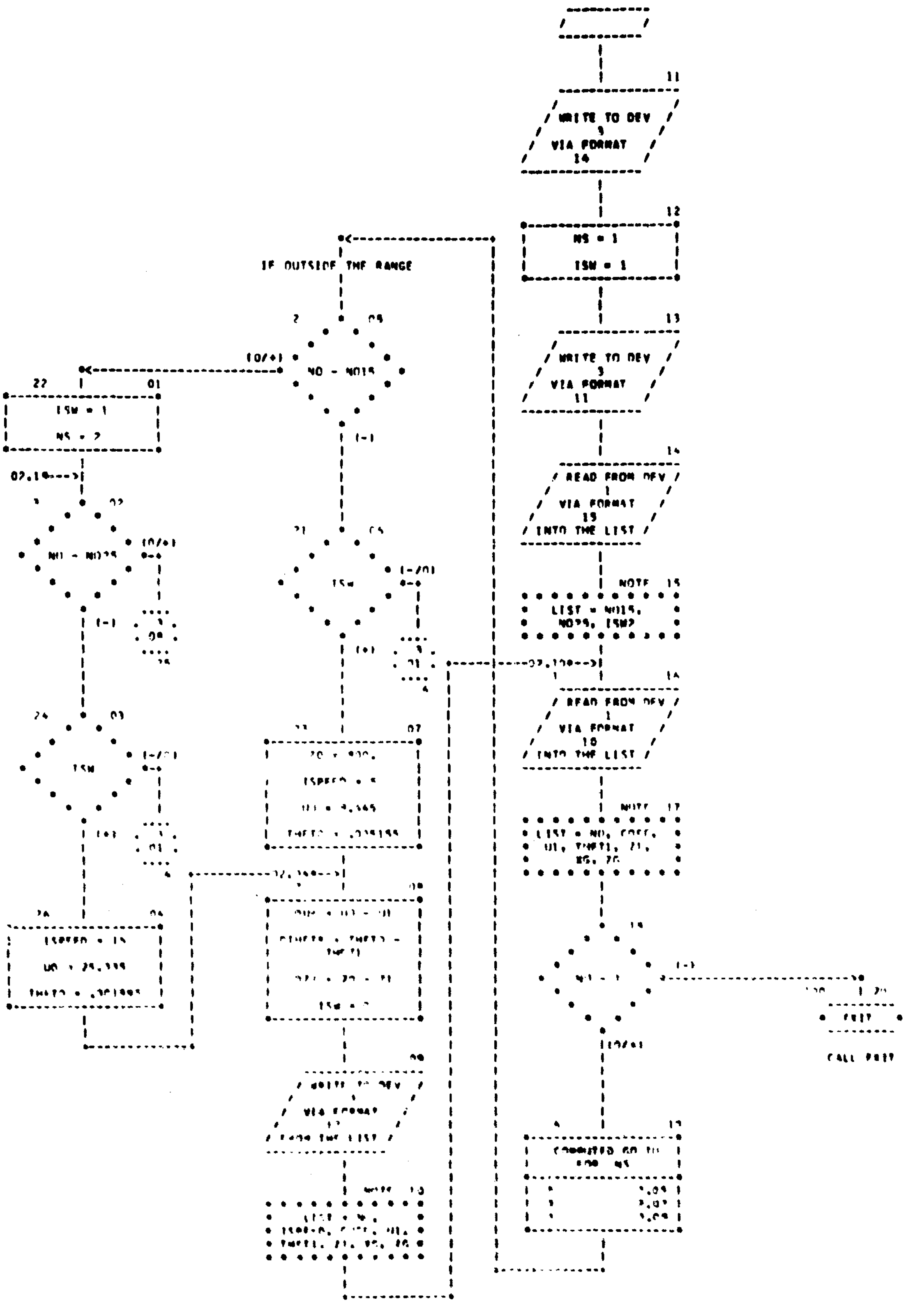


```
//        JOB    ZC691
//        EXEC FFORTRAN
C
C ZC691, ERROR CALCULATOR, DS CONTROL
C
      DIMENSION COEF(2)
      WRITE(3,14)
   14 FORMAT(1H1,T5,'ZC691, ERROR CALCULATIONS, DS CONTROL')
      NS=1
      ISW=1
      WRITE(3,11)
   11 FORMAT(1H0,T3,'NO',T7,'SPEED',T14,'COEF',T28,'PERCENT CHANGE OF VA
     1RIABLE',T93,'INPUT DATA', / T7,'(KTS)',T31,'DU',T37,'DTHETA',T47,'
     2DZ',T74,'U1',T85,'THET1', T97,'Z1',T110,'XG',T121,'ZG'//)
      READ(1,15)NO15,NO25,ISW2
   15 FORMAT(I4,6X,I4,6X,I1)
    1 READ(1,10)NO,COEF,U1,THET1,Z1,XG,ZG
   10 FORMAT(I4,6X,2A4,2X,5F10.5)
      IF(NO-1)100,6,6
    6 GO TO (2,3,30),NS
    2 IF(NO-NO15)21,22,22
   21 IF(ISW)4,4,23
   23 Z0=800.
      ISPEED=5
      U0=8.445
      THETO=.005155
    7 DUR=U0-U1
      DTHETR=THETO-THET1
      DZR=Z0-Z1
      ISW=0
      WRITE(3,12)NO,ISPEED,COEF,U1,THET1,Z1,XG,ZG
   12 FORMAT(1H0,T2,I4,T9,I2,T14,2A4,T70,F9.5,T83,F9.6,T93,F9.3,T109,F5.
     12,T120,F6.4)
      GO TO 1
    4 DU=U0-U1
      DTHET=THETO-THET1
      DZ=Z0-Z1
      DELU=100*(DUR-DU)/DUR
      DELTH=100.*(DTHETR-DTHET)/DTHETR
      DELZ=100.*(DZR-DZ)/DZR
      WRITE(3,13)NO,ISPEED,COEF,DELU,DELTH,DELZ,U1,THET1,Z1,XG,ZG
   13 FORMAT(T2,I4,T9,I2,T14,2A4,T30,F6.1,T38,F6.1,T46,F6.1,T70,F9.5,T83
     1,F9.6,T93,F9.3,T109,F5.2,T120,F6.4)
      IF(ISW2)28,29,28
   28 ISW=1
   29 GO TO 1
   22 ISW=1
      NS=2
    3 IF(NO-NO25)24,25,25
   24 IF(ISW)4,4,26
   26 ISPEED=15
      U0=25.335
      THETO=.001895
      GO TO 7
```

```
   25 ISW=1
      NS=3
   30 IF(ISW)4,4,27
   27 ISPEED=25
      U0=42.25
      THETO=.001803
      GO TO 7
  100 CALL EXIT
      END
/*
/&
```

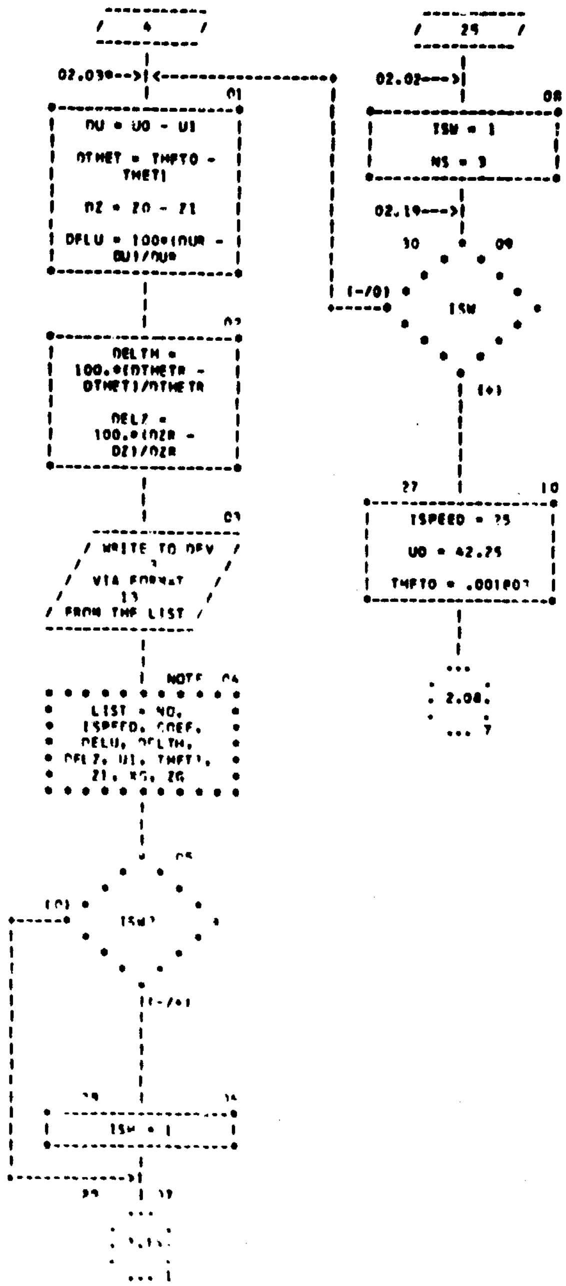
03/11/69 AUTOFLOW CHART SET - ZC691 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES



03/11/69 AUTOFLOW CHART SET - ZC691 NAVTRADEVCEN 69-C-0050-2

CHART TITLE - PROCEDURES



```
//        JOB   EC140
//        EXEC FFORTRAN
      COMMON  A(5,5,25), AA(25),NP11
      DOUBLE PRECISION  A, AA, C(3), A1,A2,A3, ROOTR(10), ROOTI(10)
      DIMENSION ISW(10)
      EQUIVALENCE (C(1),A1),(C(2),A2),(C(3),A3)
      IN=1
      IOUT=3
      TWOPI=6.28318
      LINSPP=54
    1 CONTINUE
      READ(IN,100)ZWD, ZW, ZQD, ZQ, AMWD, AMW, AMQD, AMQ
      IF(ZWD-999.)10,90,90
   10 CONTINUE
      READ(IN,100) AW, AQ, AMAWQ, AMWAW,  ZWAQ, ZWAW
      READ(IN,100) XUD,A11,A12,ZSTR,AMSTR,XQQ,XWQ
      READ(IN,100) AL, AM, AIYY, B, ZB,RHO
      IPAGE=1
      WRITE(IOUT, 2)  IPAGE
    2 FORMAT(1H1,55X,'EC140',50X,'PAGE',I6/)
      WRITE(IOUT,4)ZWD, ZW, ZQD, ZQ, AMWD, AMW, AMQD, AMQ,
     1 AW, AQ, AMAWQ, AMWAW, ZWAQ, ZWAW,
     AXUD, A11, A12, ZSTR, AMSTR, XQQ, XWQ,
     2 AL, AM, AIYY, B, ZB, RHO
    4 FORMAT(1H ,8X,'ZWD',14X,'ZW',13X,'ZQD',14X,'ZQ',13X,'MWD',14X,
     1'MW',13X,'MQD',14X,'MQ'/
     21H ,8E16.6//
     A1H ,8X,'AW',15X,'AQ',13X,'MAWQ',13X,'MWAW',11X,'ZWAQ',13X,'ZWAW'/
     B1H ,6E16.6//
     C1H ,8X,'XUD',14X,'A11',12X,'A12',14X,'ZSTR',11X,'MSTR',13X,'XQQ',
     D12X,'XWQ'/1H ,7E16.6//
     31H ,10X,'L',15X,'M',14X,'IY',15X,'B',14X,'ZB',13X,'RHO'/
     41H ,6E16.6/)
      ZWH = ZW
      AMWH = AMW
      AMQH = AMQ
      WRITE(IOUT,6)
    6 FORMAT(1H ,9X,'MW',14X,'MQ',14X,'ZW'/
     1        1H ,9X,'A1',14X,'A2',14X,'A3',14X,'A4',14X,'A5',15X,'U'/
     21H ,10X,'R',15X,'I',15X,'R',15X,'I',15X,'R',15X,'I',15X,
     A'R',15X,'I'/
     31H ,9X,'WN',12X,'T(1/2)',13X,'P',15X,'D'/)
      LINS=16
   50 CONTINUE
      READ(IN,100) U
  100 FORMAT(8F10.5)
      IF(U) 60,1,60
   60 CONTINUE
      U = U * 1.689
      WBAR = AW/U
      QBAR = AQ*AL/U
      AMW = AMWH +AMWAW*WBAR
      AMQ = AMQH +AMAWQ*WBAR
      ZW = ZWH +ZWAQ*QBAR+ZWAW*WBAR
```

```
      N = 2
      NCOL = 3
      M = N*NCOL+1
      DO 62 I = 1,NCOL
      DO 62 J = 1,NCOL
      DO 62 K = 1,M
   62 A(I,J,K) = 0.
      AL2=AL*AL*RHO/2.
      AL3=AL2*AL
      AL4=AL3*AL
      AL5=AL4*AL
      T1=AL3*ZWD-AM
      T2=AL4*AMWD
      T3=B*ZB
      T4=U*AM+AL3*U*ZQ
      T5=AL2*U*ZW
      T6=AL4*U*AMQ
      T7=AL5*AMQD-AIYY
      T8=AL3*U*AMW
      T9=AL4*ZQD
      A1 = T7
      A2 = T6
      A3 = T3
      CALL PLACE(1,1,C)
      A1 = T2
      A2 = T8
      A3 = 0.
      CALL PLACE(1,2,C)
      A1=0.
      A2=T9
      A3= T4
      CALL PLACE(2,1,C)
      A2 = T1
      A3 = T5
      CALL PLACE(2,2,C)
      A2 = 0.
      A3 = AL3*U*(XQQ*QBAR+XWQ*WBAR)-AM*WBAR*U
      CALL PLACE(3,1,C)
      A3 = 0.
      CALL PLACE(3,2,C)
      A2 = AL3*XUD-AM
      A3 = AL2*U*(A11+A11+A12)
      CALL PLACE(3,3,C)
      A2 = AL3*AMSTR*2.*U
      A3 =  0.
      CALL PLACE(1,3,C)
      A2 = 0.
      A3 =  AL3*ZSTR*2.*U
      CALL PLACE(2,3,C)
      CALL CHRFQN
      NP1=NP11+1
      CALL MULLER(AA, NP11,ROOTR,ROOTI,ISW,IERR)
      DO 74 I = 1, NP11
      IF(DABS(ROOTI(I))-1.D-5)74,74,76
```

```
74 CONTINUE
   WRITE(IOUT,75)
75 FORMAT(1H ,'NO ROOT WITH IMAGINARY PART FOUND')
   GO TO 50
76 TR = DABS(ROOTR(I))
   TI = DABS(ROOTI(I))
   WN = SQRT(TR*TR+TI*TI)
   THALF = .693/TR
   P = TWOPI/TI
   D = TR/WN
   IF(LINS-LINSPP)80,79,79
79 IPAGE=IPAGE+1
   WRITE(IOUT,2)    IPAGE
   WRITE(IOUT,6)
   LINS=6
80 CONTINUE
   WRITE (IOUT,82) AMW,AMO, ZW
   WRITE (IOUT,82) (AA(I),I=1,NP1),U
   WRITE (IOUT,82) (ROOTR(I),ROOTI(I),I=1,NP11)
   WRITE(IOUT,82) WN,THALF,P,D
   WRITE(IOUT,83)
82 FORMAT(8F16.6)
83 FORMAT(1H/)
   LINS=LINS+5
   GO TO 50
90 CALL EXIT
   END
```

```
      SUBROUTINE PLACE(I,J,X)
      DOUBLE PRECISION  A(5,5,25), X(1)
      COMMON A
      KK = 3
      DO 400 K=1,3
      A(I,J,KK) = X(K)
  400 KK=KK-1
      RETURN
      END
```

```
      SUBROUTINE CHREQN
      DOUBLE PRECISION COE,C1,C2,AA,A,SB,SA
      DIMENSION A(5,5,25),COE(25),MAT(8,8),C1(25),C2(25),AA(25)
      COMMON A, AA, NP11
      N = 2
      NCOL = 3
    1 M=N*NCOL+1
 1501 N1=N+1
      NCOL2= NCOL*NCOL
      L1 = N1 + 1
      DO 303 I=1,M
      C1(I)=0.
      AA(I)=0.
  303 C2(I) = 0.
      NP1=N+1
C     FIND DEGREE OF EACH MATRIX ELEMENT
      DO  2   I=1,NCOL
      DO  2   J=1,NCOL
      MAT(I,J)=0
      DO  2   K=1,N1
      IF(A(I,J,K)) 600,2,600
  600 MAT(I,J)=K
    2 CONTINUE
C     TRIANGULARIZE THE MATRIX
      J3=0
      J1=1
   10 J9=0
      DO   3 I=J1,NCOL
      IF(MAT(I,J1))100,601,601
  601 IF(MAT(I,J1))602,3,602
  602 J9=J9+1
      J3=I
    3 CONTINUE
C     J1 = COLUMN NUMBER
C     J9 = NUMBER OF NON-ZERO ELEMENTS IN THIS COLUMN
C     J3 = LAST NON-ZERO ELEMENT IN THIS COLUMN
   11 IF(J9-1)100,603,12
  603 IF(J3-J1)100,112,204
  204 DO 4   J=J1,NCOL
      J2= MAX0(MAT(J3,J),MAT(J1,J))
      J4=MAT(J3,J)
      MAT(J3,J)=MAT(J1,J)
      MAT(J1,J)= J4
      DO   4 K=1,J2
      SA=A(J3,J,K)
      A(J3,J,K)=A(J1,J,K)
      A(J1,J,K)=-SA
    4 CONTINUE
      GO TO 112
   12 J3=J1+1
      DO 111 I=J3,NCOL
   13 IF(MAT(I,J1))100,111,205
  205 IF(MAT(J1,J1))100,14,206
  206 IF(MAT(I,J1)-MAT(J1,J1))14,15,15
```

```
C       INTERCHANGE ROW I WITH J1
   14 DO  6 J= J1,NCOL
      J2= MAX0(MAT(J1,J),MAT(I,J))
      J4= MAT(J1,J)
      MAT(J1,J)=MAT(I,J)
      MAT(I,J)=J4
      DO 6 K=1,J2
      SA= A(I,J,K)
      A(I,J,K)=A(J1,J,K)
      A(J1,J,K)=-SA
    6 CONTINUE
      GO TO 13
C
   15 J7=MAT(I,J1)
      J5=MAT(J1,J1)
      J6=J7-J5
      SB=A(I,J1,J7)/A(J1,J1,J5)
      IF(ABS(SB)-4.)16,207,207
  207 IF(J6)100,14,16
   16 DO 19   J= J1,NCOL
      J5=MAT(J1,J)
      DO 19   K=1,J5
      J7= K+J6
      IF(J7-M)17,17,110
   17 IF(ABS(A(I,J,J7)-SB*A(J1,J,K))-2.E-15)18,18,208
  208 A(I,J,J7) = A(I,J,J7)-SB*A(J1,J,K)
      GO TO 19
   18 A(I,J,J7)=0.
   19 CONTINUE
  110 DO   7 J=J1,NCOL
      J7= MAX0(MAT(I,J),MAT(J1,J)+J6)
      MAT(I,J)=0
      DO   7  K=1,J7
      IF(A(I,J,K))209,7,209
  209 MAT(I,J)=K
    7 CONTINUE
  111 CONTINUE
      GO TO 10
  112 J1=J1+1
      IF(J1-NCOL)10,210,210
C       GET PRODUCT OF DIAGONAL ELEMENTS
  210 DO 115 J=1,NCOL
      J2=MAT(J,J)
      DO 8  K=1,J2
    8 C1(K)=A(J,J,K)
  113 IF(J-1)100,114,211
  211 DO 9  K=1,NP1
    9 C2(K)=COE(K)
      DO 116  K=1,M
  116 COE(K)=0.
      IF(J2)100,115,212
  212 DO 117 K=1,J2
      DO 117 J10=1,NP1
      J11= K+J10-1
```

```
      COE(J11)=COE(J11)+C1(K)*C2(J10)
  117 CONTINUE
      NP1=J11
      GO TO 115
  114 DO 118 K=1,J2
  118 COE(K)=C1(K)
      NP1=J2
  115 CONTINUF
  100 II=NP1
      DO 401 I=1,NP1
       AA(II)=COE(I)
  401 II=II-1
      NP11=NP1-1
      RETURN
      END
```

```
      SUBROUTINE MULLER(COE,N1,ROOTR,ROOTI,ISW,IERR)
C     MULLER ROUTINE FOR ZEROES OF POLYNOMIALS WITH REAL COEFFICIENTS
C
C     COE IS THE ARRAY OF POLYNOMIAL COEFFICIENTS ORDERED FROM HIGHEST
C     TO LOWEST POWER OF X
C     N1 IS THE DEGREE OF THE POLYNOMIAL
C     ROOTR IS THE ARRAY OF REAL COMPONENTS OF THE ROOTS
C     ROOTI IS THE ARRAY OF IMAGINARY COMPONENTS OF THE ROOTS
C     ISW IS AN ARRAY DEFINING THE VALIDITY OF THE ROOTS
C              ISW(N) = 0     THE NTH ROOT HAS BEEN STORED IN ROOTR(N) AND
C                             ROOTI(N)
C              ISW(N) = 1     THE NTH ROOT HAS BEEN STORED IN ROOTR(N) AND
C                             ROOTI(N), BUT IT MAY NOT BE VALID
C     IERR IS AN ERROR CODE WHICH HAS THE FOLLOWING SIGNIFICANCE.
C              IERR = 0       ALL ROOTS FOUND CORRECTLY
C              IERR = 1       ONE OR MORE ROOTS MAY BE INVALID.  TEST THE
C                             ISW ARRAY .
C              IERR = 2       POLYNOMIAL DEGREE IS LESS THAN 1
C              IERR = 3       POLYNOMIAL DEGREE IS LESS THAN N1
C
C     FOR A POLYNOMIAL OF DEGREE N1 THE COE ARRAY SHOULD BE DIMENSIONED
C     N1+ 1 IN THE USER PROGRAM.  THE OTHER ARRAYS SHOULD BE DIMENSIONED
C     N1 IN THE USER PROGRAM.
C     THE POLYNOMIAL IS SCALED TO AVOID ARITHMETIC OVERFLOW.  ALL
C     SCALING USES FACTORS OF 16.  TO CHANGE TO FACTORS OF X, SET BASE
C     = X   AND   CONST = LN(X)  IN SUBROUTINE
C     THIS SUBROUTINE USES DOUBLE PRECISION ARITHMETIC. SINGLE PRECISION
C     IS NOT RECOMMENDED.
C
      DIMENSION COE(1), ROOTR(1), ROOTI(1), ISW(1)
      DOUBLE PRECISION COE,TE2,TE3,DIV,TEE7,DE15,TE13,HELL,TEM2,ALP1R,
     1ALP2R,ALP3R,TEST1,7TAU2,AXR,TE5,TE4,UPP,TEMR,DE16,TE14,BELL,ROOTR,
     2ALP1I,ALP2I,ALP3I,TEST2,AXI,TE6,TE7,TE8,TEMI,TE11,TE15,TAU2,ROOTI,
     3BET1R,BET2R,BET3R,ALP4R,TE1,TEM,TE9,TAU,TE10,TE12,TE16,TEM1,Z1,
     4BET1I,BET2I,BET3I,ALP4I,Z2,D1,D2,FACTOR
C
      BASE = 16.
      CONST = 2.77259
C
      CALL MASK(0)
      IF(N1 - 1) 27,28,28
   27 IERR = 2
      GO TO 193
   28 IERR = 0
      FACTOR = 0.
      N2=N1+1
      N4=0
      I=N1+1
   19 IF(COE(I))9,7,9
    7 N4=N4+1
      ROOTR(N4)=0.
      ROOTI(N4)=0.
      I=I-1
      IF(N4-N1)19,37,19
```

```
  9 CONTINUE
    IF(COE(1)) 190,192,190
190 TEMP = DABS(COE(I)/COE(1))
    TEMP = ALOG(TEMP)/CONST/(I-1)
    K2 = TEMP + SIGN(.5,TEMP)
    TEMP = DABS(COE(1))
    TEMP = ALOG(TEMP)/CONST
    K1 = TEMP + SIGN(.5,TEMP)
    DO 191 I = 1,N2
191 COE(I) = COE(I)/BASE**(K1 + K2*(I-1))
    FACTOR = BASE**K2
    GO TO 10
192 IERR = 3
193 DO 194 I = 1,N1
    ROOTR(I) = 0.
    ROOTI(I) = 0.
194 ISW(I) = 1
    RETURN
 10 AXR=0.8
    AXI=0.
    L=1
    N3=1
    ALP1R=AXR
    ALP1I=AXI
    M=1
    GO TO 99
 11 BET1R=TEMR
    BET1I=TEMI
    AXR=0.85
    ALP2R=AXR
    ALP2I=AXI
    M=2
    GO TO 99
 12 BET2R=TEMR
    BET2I=TEMI
    AXR=0.9
    ALP3R=AXR
    ALP3I=AXI
    M=3
    GO TO 99
 13 BET3R=TEMR
    BET3I=TEMI
 14 TE1=ALP1R-ALP3R
    TE2=ALP1I-ALP3I
    TE5=ALP3R-ALP2R
    TE6=ALP3I-ALP2I
    TEM=TE5*TE5+TE6*TE6
    TE3=(TE1*TE5+TE2*TE6)/TEM
    TE4=(TE2*TE5-TE1*TE6)/TEM
    TE7=TE3+1.
    TE9=TE3*TE3-TE4*TE4
    TE10=2.*TE3*TE4
    DE15=TE7*BET3R-TE4*BET3I
    DE16=TE7*BET3I+TE4*BET3R
```

```
      TE11=TE3*BET2R-TE4*BET2I+BET1R-DE15
      TE12=TE3*BET2I+TE4*BET2R+BET1I-DE16
      TE7=TE9-1.
      TE1=TE9*BET2R-TE10*BET2I
      TE2=TE9*BET2I+TE10*BET2R
      TE13=TE1-BET1R-TE7*BET3R+TE10*BET3I
      TE14=TE2-BET1I-TE7*BET3I-TE10*BET3R
      TE15=DE15*TE3-DE16*TE4
      TE16=DE15*TE4+DE16*TE3
      TE1=TE13*TE13-TE14*TE14-4.*(TE11*TE15-TE12*TE16)
      TE2=2.*TE13*TE14-4.*(TE12*TE15+TE11*TE16)
      TEST1=DABS(TE1)
      TEST2=DABS(TE2)
      IF(TEST1-TEST2) 300,301,301
  300 DIV=TEST2
      UPP=TEST1
      GO TO 303
  301 DIV=TEST1
      UPP=TEST2
      IF(DIV-1.D-70) 999,303,303
  999 DIV= 1.D-70
  303 TEM=DIV*DSQRT(1.+(UPP/DIV)*(UPP/DIV))
      IF(TE1)113,113,112
  113 TE4=DSQRT(.5*(TEM-TE1))
      TE3=.5*TE2/TE4
      GO TO 111
  112 TE3=DSQRT(.5*(TEM+TE1))
      IF(TE2)110,200,200
  110 TE3=-TE3
  200 TE4=.5*TE2/TE3
  111 TE7=TE13+TE3
      TE8=TE14+TE4
      TE9=TE13-TE3
      TE10=TE14-TE4
      TE1=2.*TE15
      TE2=2.*TE16
      IF(TE7*TE7+TE8*TE8-TE9*TE9-TE10*TE10)204,204,205
  204 TE7=TE9
      TE8=TE10
  205 TEM=TE7*TE7+TE8*TE8
      IF(TEM -1.D-70) 998,997,997
  998 TEM= 1.D-70
  997 TE3=(TE1*TE7 + TE2 *TE8)/TEM
      TE4=(TE2*TE7-TE1*TE8)/TEM
      AXR=ALP3R+TE3*TE5-TE4*TE6
      AXI=ALP3I+TE3*TE6+TE4*TE5
      ALP4R=AXR
      ALP4I=AXI
      M=4
      GO TO 99
   15 N6=1
   38 O1=DABS(HELL)+DABS(RELL)
      TE7=DABS(ALP3R-AXR)+DABS(ALP3I-AXI)
      TEE7=   DABS(AXR)+DABS(AXI)
```

```
C IS THE FUNCTION VALUE NEAR ZERO ?
      IF(Q1    -1.D-20                        )161,161,      16
C IS THE ROOT SMALL ?
161   IF(TEE7-1.0D-03)162,16,16
C IS THE CURRENT ESTIMATE FOR THE ROOT ESSENTIALLY
C THE SAME AS THE PREVIOUS ESTIMATE ?
162   IF(TE7-1.0D-12) 18,17,17
C ARE THE CURRENT AND PREVIOUS ESTIMATES OF THE ROOT ESSENTIALLY
C THE SAME WHEN COMPARED TO THE MAGNITUDE OF THE ROOT ?
16     Q2=TE7/ TEE7
      IF(Q2 - 1.E-7)18,18,17
   17 N3=N3+1
      ALP1R=ALP2R
      ALP1I=ALP2I
      ALP2R=ALP3R
      ALP2I=ALP3I
      ALP3R=ALP4R
      ALP3I=ALP4I
      BET1R=BET2R
      BET1I=BET2I
      BET2R=BET3R
      BET2I=BET3I
      BET3R=TEMR
      BET3I=TEMI
      IF(N3-200)14,25,25
   25 ISWT = 1
      GO TO 26
   18 ISWT = 0
   26 N4 = N4 + 1
      ISW(N4) = ISWT
      ROOTR(N4)=ALP4R
      ROOTI(N4)=ALP4I
      N3=0
   41 IF(N4-N1)30,37,37
   37 CONTINUE
      IF(FACTOR) 140,140,138
  138 DO 139 I = 1,N1
      ROOTR(I) = ROOTR(I)*FACTOR
      ROOTI(I) = ROOTI(I)*FACTOR
  139 COE(I) = COE(I)*BASE**(K1+K2*(I-1))
      COE(N2) = COE(N2)*BASE**(K1+K2*N1)
  140 DO 141 I=1,N1
      IF(ISW(I)) 141,141,142
  141 CONTINUE
      IERR = 0
      GO TO 3001
  142 IERR = 1
 3001 RETURN
   30 IF(DABS(ROOTI(N4)/ROOTR(N4)) - 1.E-5) 10,10,131
  131 IF(ISWT) 31,132,31
  132 GO TO (133,134),L
  133 N4 = N4 + 1
      M4 = N4
      GO TO 135
```

```
  134 M4 = N4 - 1
  135 ROOTR(M4) = AXR
      ROOTI(M4) = -AXI
      ISW(M4) = 0
      IF(M4 - N1) 10,37,37
   31 GO TO(32,10),L
   32 AXR=ALP1R
      AXI=-ALP1I
      ALP1I=-ALP1I
      M=5
      GO TO 99
   33 BET1R=TEMR
      BET1I=TEMI
      AXR=ALP2R
      AXI=-ALP2I
      ALP2I=-ALP2I
      M=6
      GO TO 99
   34 BET2R=TEMR
      BET2I=TEMI
      AXR=ALP3R
      AXI=-ALP3I
      ALP3I=-ALP3I
      L=2
      M=3
   99 TEMR=COE(1)
      TEMI=0.0
      DO 100 I=1,N1
      TE1=TEMR*AXR-TEMI*AXI
      TEMI=TEMI*AXR+TEMR*AXI
100   TEMR=      TE1+COE(I+1)
      HELL=TEMR
      BELL=TEMI
   42 IF(N4)102,103,102
  102 DO 101 I=1,N4
      TEM1=AXR-ROOTR(I)
      TEM2=AXI-ROOTI(I)
      TE1=TEM1*TEM1+TEM2*TEM2
      TE2=(TEMR*TEM1+TEMI*TEM2)/TE1
      TEMI=(TEMI*TEM1-TEMR*TEM2)/TE1
  101 TEMR=TE2
  103 GO TO(11,12,13,15,33,34),M
      END
/*
/&
```

09/11/69

AUTOFLOW CHART SET - FC140 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES

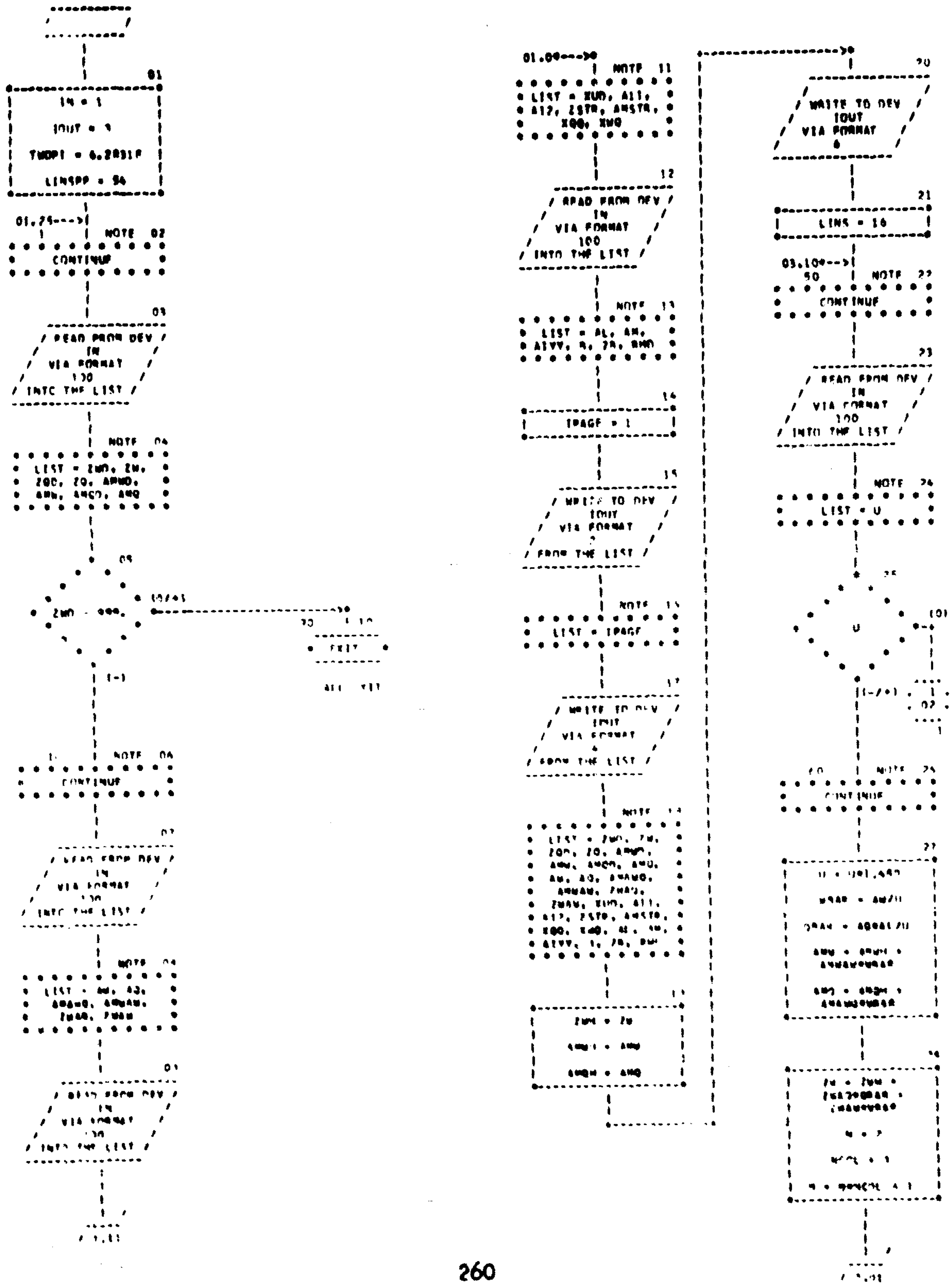


CHART TITLE - PROCEDURES

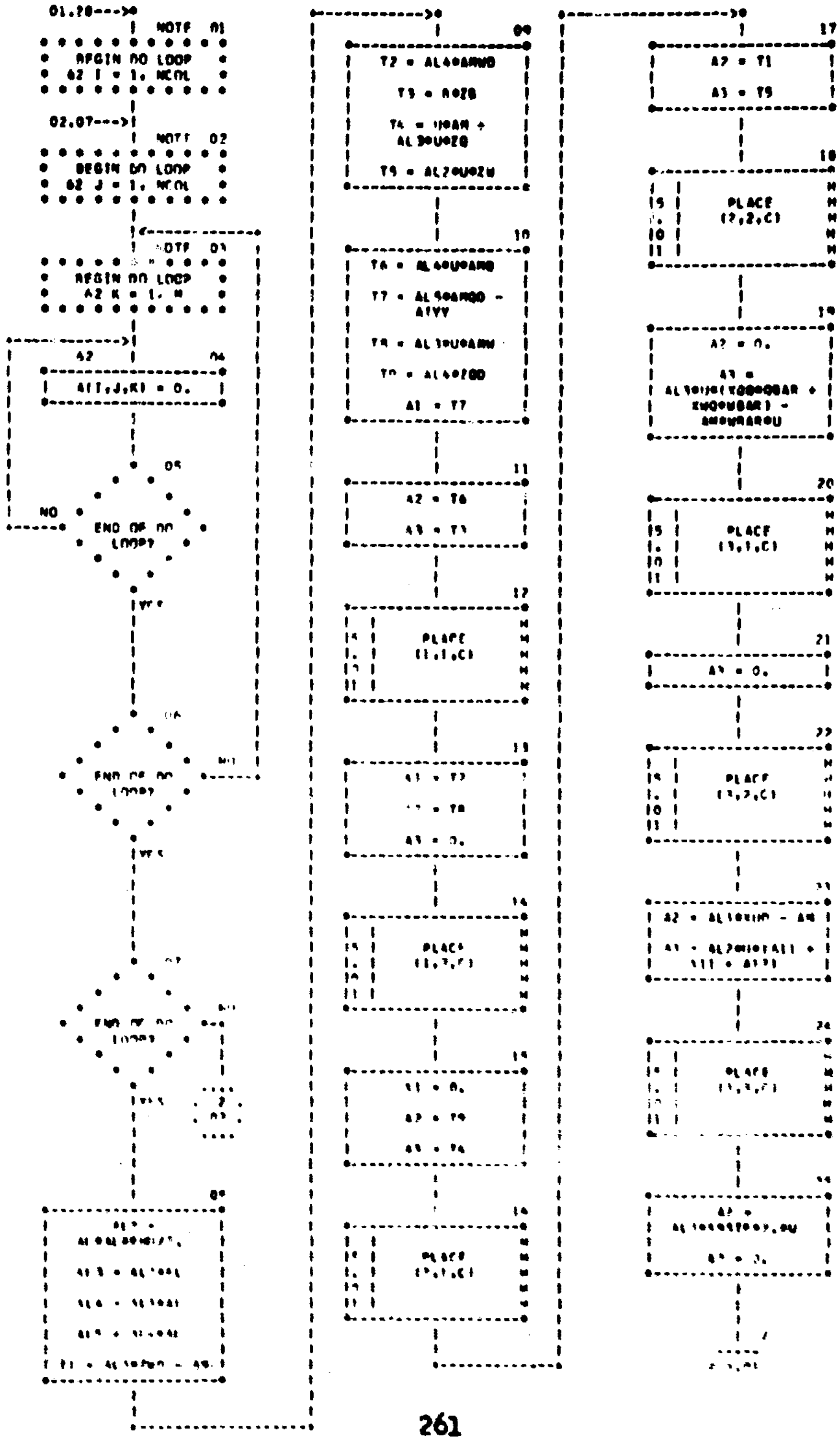


CHART TITLE - PROCEDURES

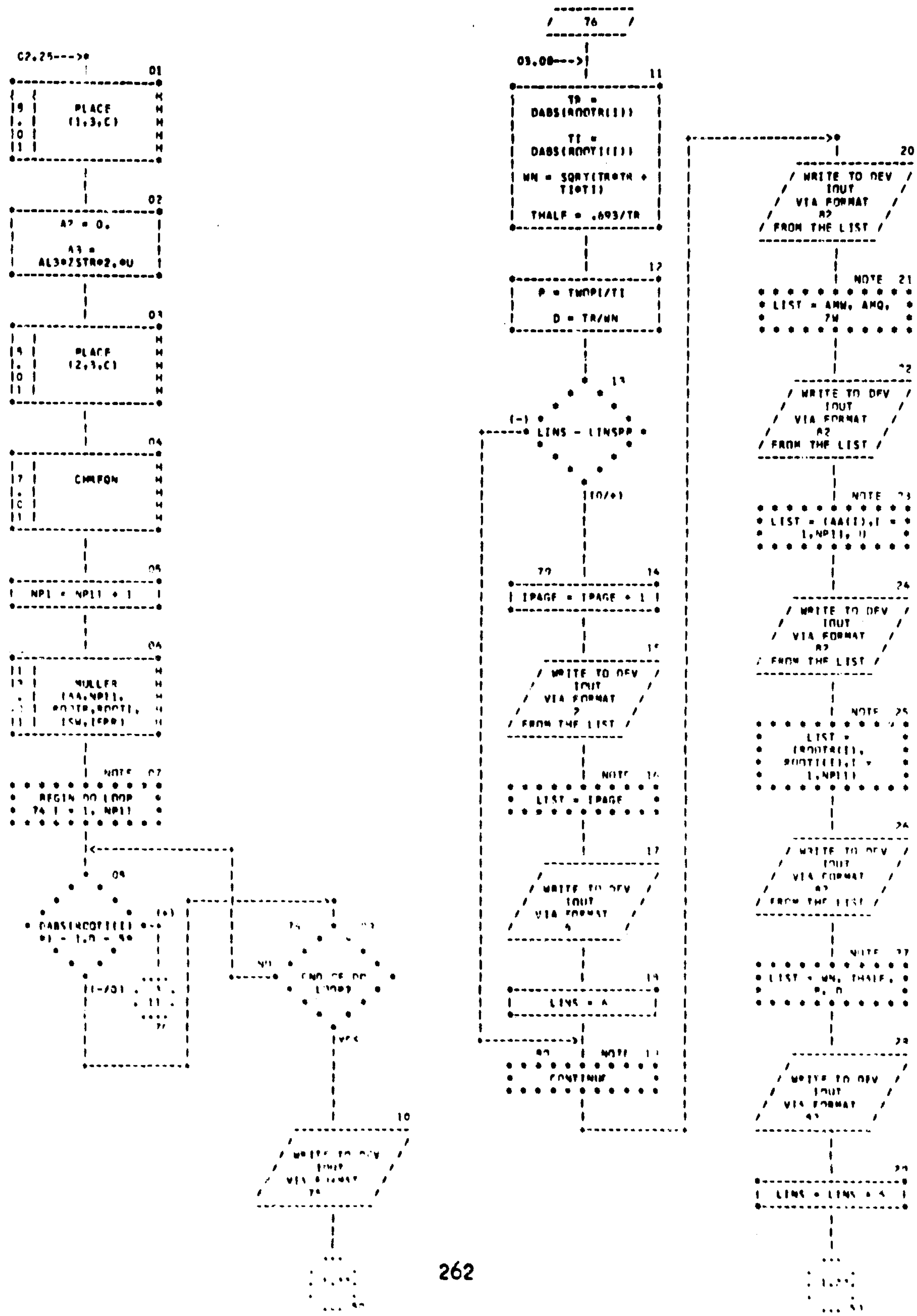


CHART TITLE - SUBROUTINE PLACE(I,J,X)

```
                    ------------
                   /  PLACE    /
                   ------------
                         |
               02,12*-->|
EC520          02,25     |           01
               *-------------------------*
               |         KK = 3          |
               *-------------------------*
                         |
                         |
                         |   NOTE   02
               *  *  *  *  *  *  *  *  *  *  *
               *    BEGIN DO LOOP        *
               *    400 K = 1, 3         *
               *  *  *  *  *  *  *  *  *  *  *
                         |
       *--------------->|
       |                 |           03
       |       *-------------------------*
       |       | A(I,J,KK) = X(K)        |
       |       *-------------------------*
       |                 |
       |                 |
       |       400       |           04
       |       *-------------------------*
       |       |     KK = KK - 1         |
       |       *-------------------------*
       |                 |
       |                 |
       |                 *     05
       |              *     *
       |           *           *
       |   NO   *                 *
       *------*    END OF DO        *
                *     LOOP?       *
                   *           *
                      *     *
                         *
                         |YES
                         |
                         |
                         |
                         |  06
                    ----------
                   *  EXIT    *
                    ----------
```

03/11/69 AUTOFLOW CHART SET - EC140 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE CHRFQN

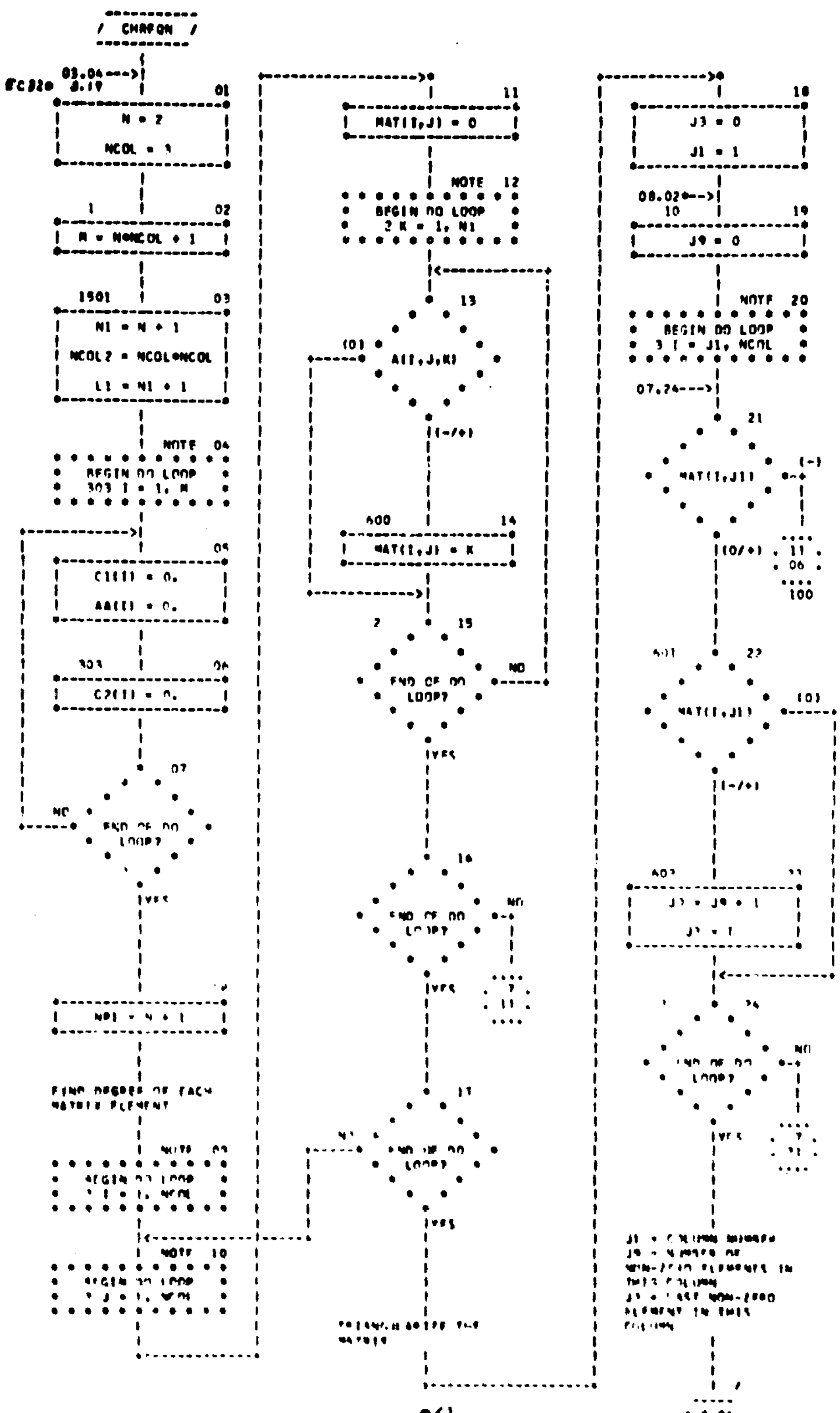


CHART TITLE - SUBROUTINE CHREON

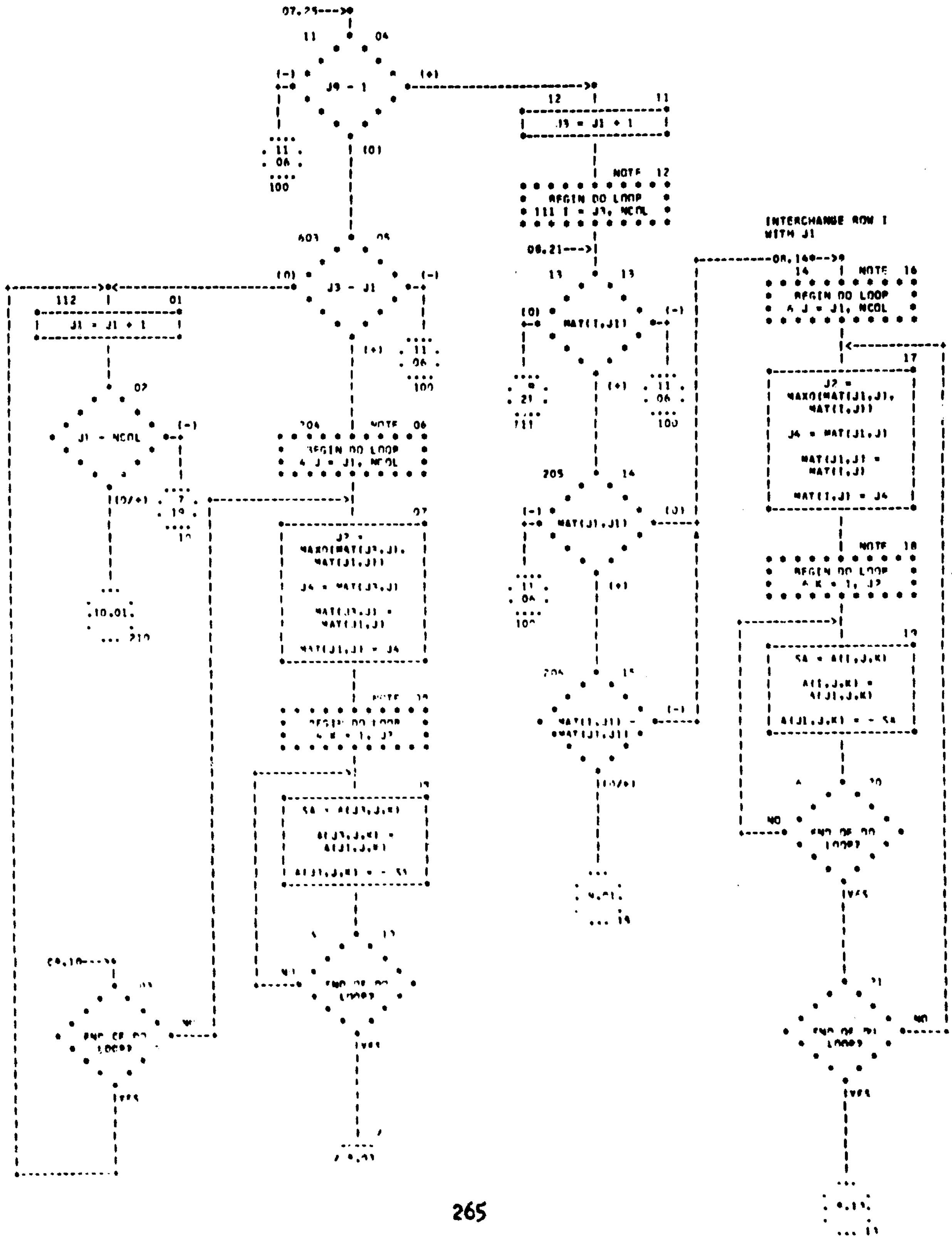
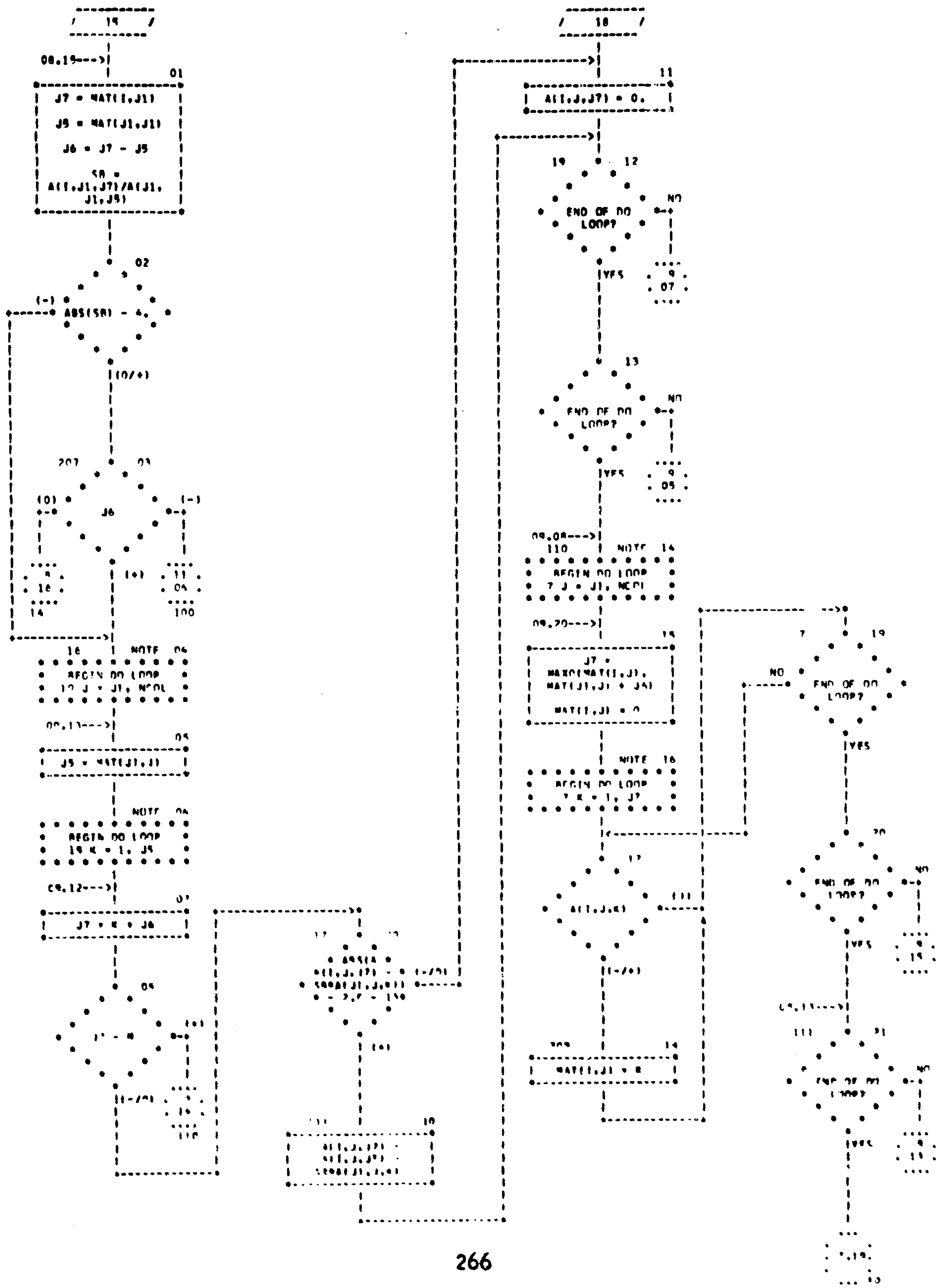


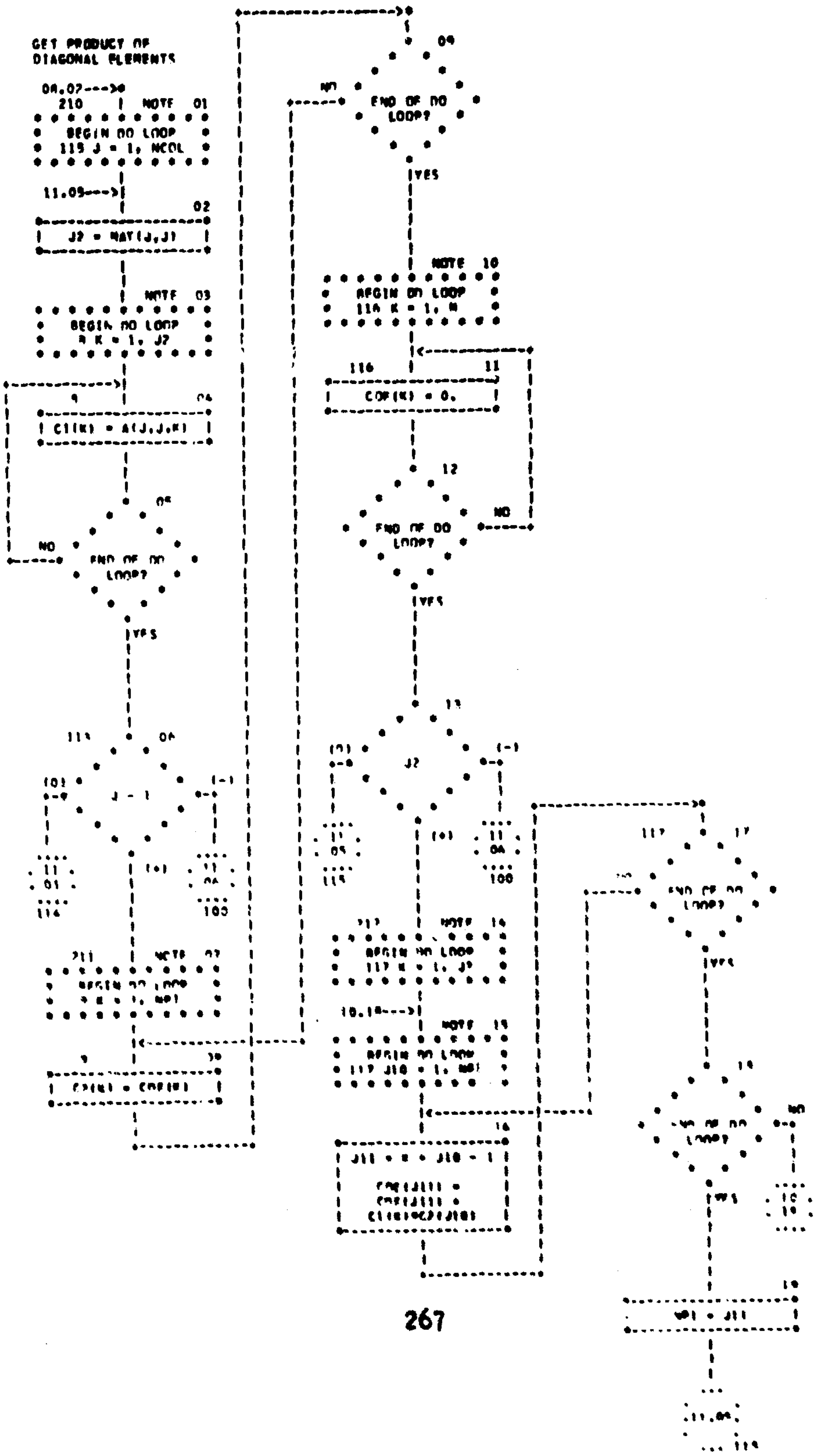
CHART TITLE - SUBROUTINE CHRFON



03/11/69

AUTOFLOW CHART SET - EC140 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE CHREGN



03/11/69

AUTOFLOW CHART SET - EC140 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE CHREQN

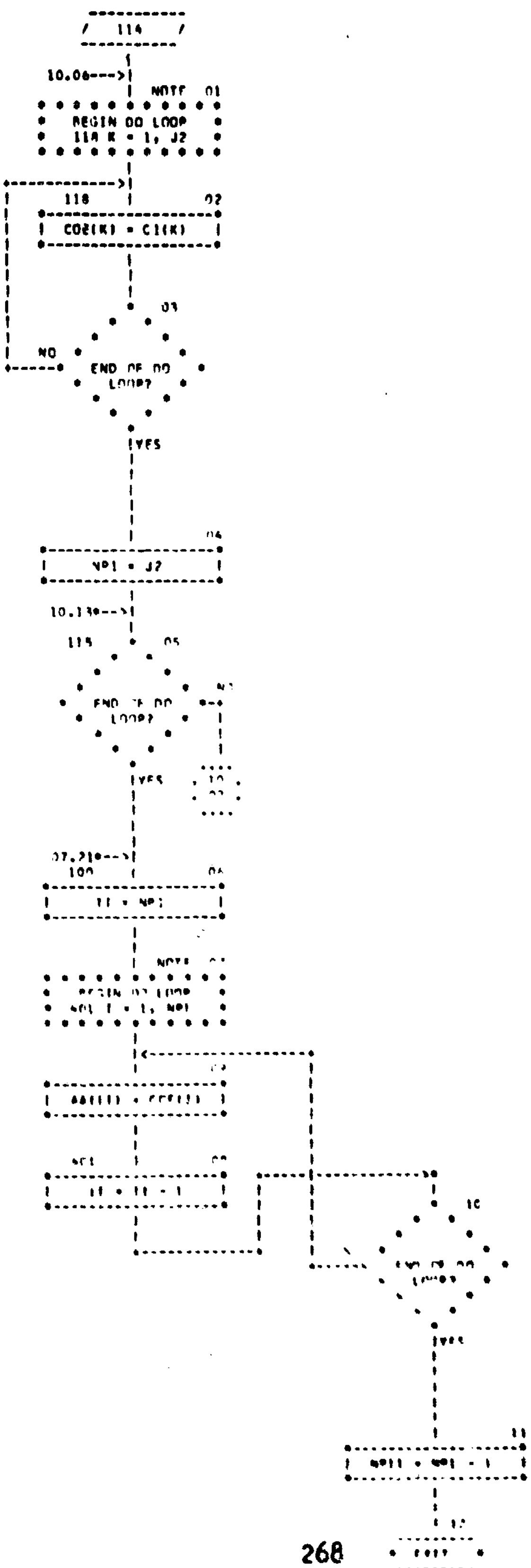


CHART TITLE - SUBROUTINE MULLER(COF,N1,ROOTR,ROOTI,ISW,IERR)

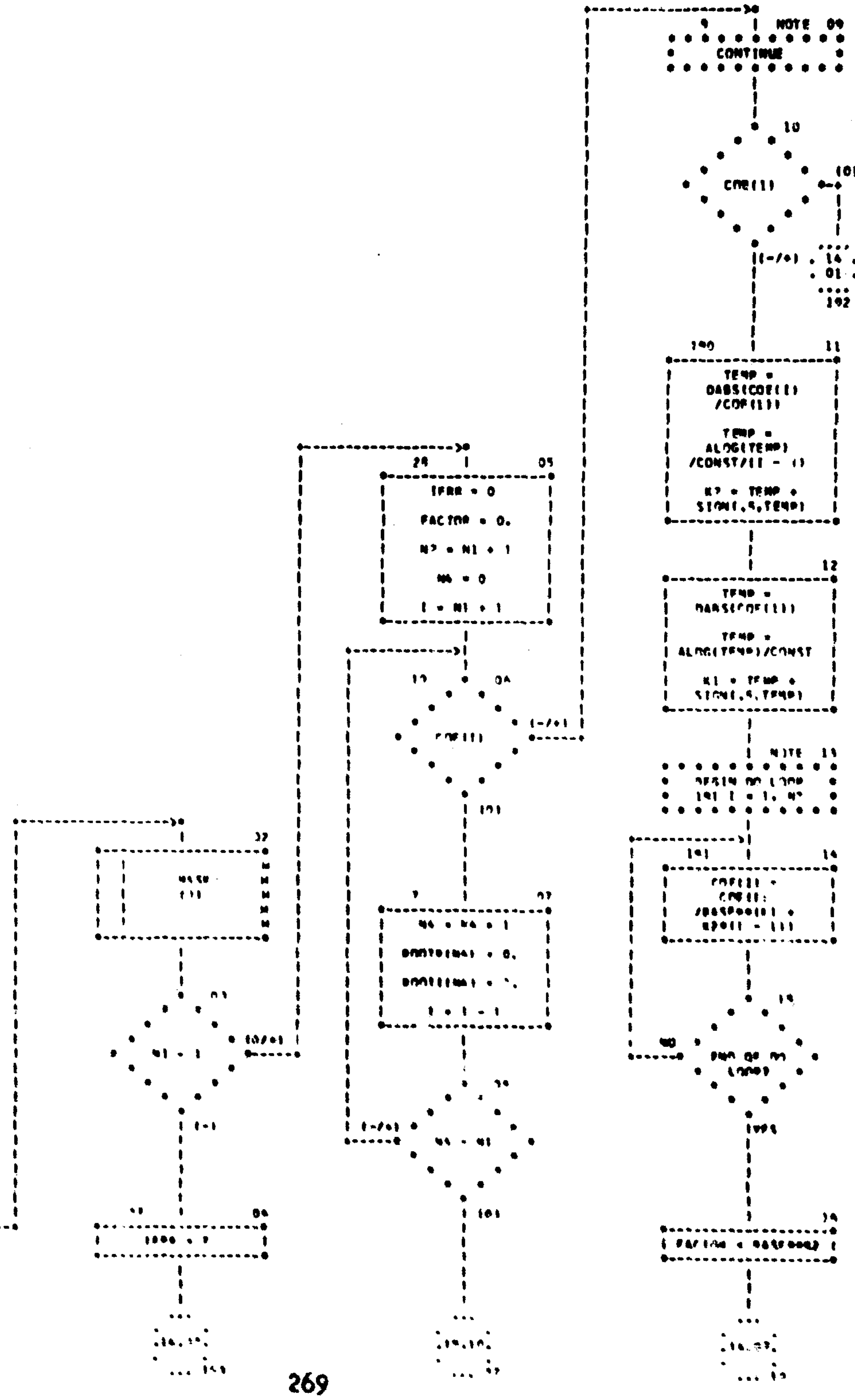
/ MULLER /

09.06---70
EC320 03.29---70
MULLER ROUTINE FOR
ZEROES OF POLYNOMIALS
WITH REAL
COEFFICIENTS

COF IS THE ARRAY OF
POLYNOMIAL
COEFFICIENTS ORDERED
FROM HIGHEST
TO LOWEST POWER OF X
N1 IS THE DEGREE OF
THE POLYNOMIAL
ROOTR IS THE ARRAY OF
REAL COMPONENTS OF
THE ROOTS
ROOT IS THE ARRAY OF
IMAGINARY COMPONENTS
OF THE ROOTS
ISW IS AN ARRAY
DEFINING THE VALIDITY
OF THE ROOTS
ISW(N) = 0 THE NTH
ROOT HAS BEEN STORED
IN ROOTR(N) AND
ROOTI(N)
ISW(N) = 1 THE NTH
ROOT HAS BEEN STORED
IN ROOTR(N) AND
ROOTI(N), BUT IT MAY
NOT BE VALID
IERR IS AN ERROR CODE
WHICH HAS THE
FOLLOWING
SIGNIFICANCE.
IERR = 0 ALL
ROOTS FOUND CORRECTLY
IERR = 1 ONE OR
MORE ROOTS MAY BE
INVALID. TEST THE
ISW ARRAY .
IERR = 2
POLYNOMIAL DEGREE IS
LESS THAN 1
IERR = 3
POLYNOMIAL DEGREE IS
LESS THAN N1

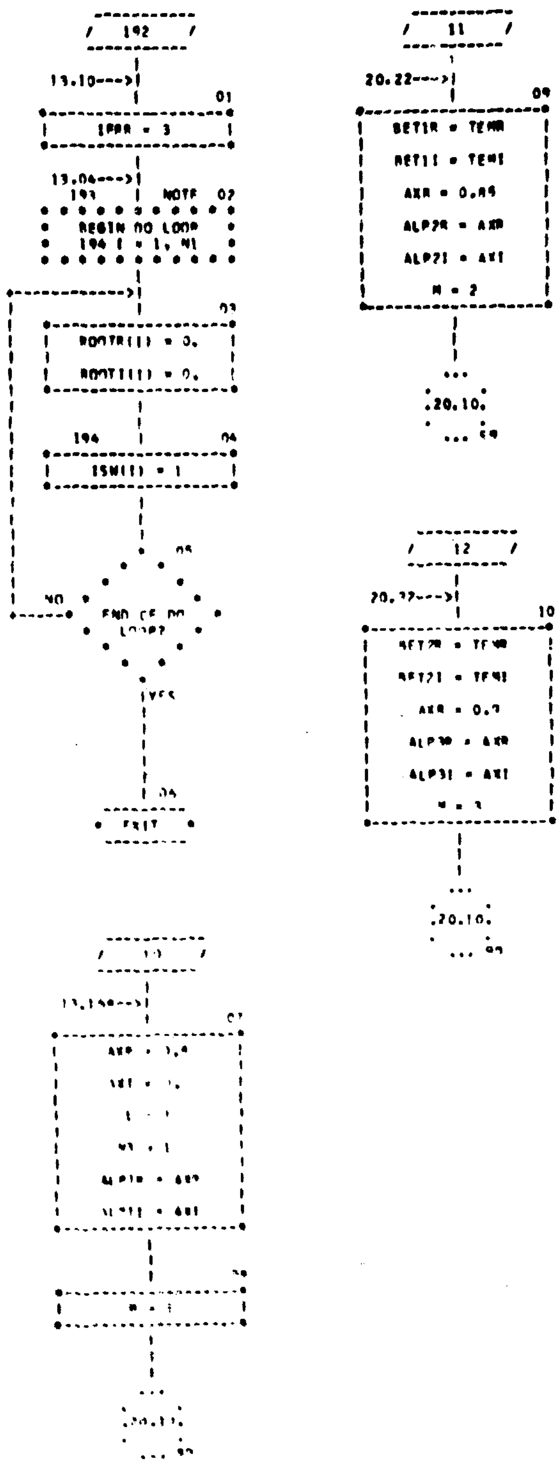
FOR A POLYNOMIAL OF
DEGREE N1 THE COF
ARRAY SHOULD BE
DIMENSIONED
N1+ 1 IN THE USER
PROGRAM. THE OTHER
ARRAYS SHOULD BE
DIMENSIONED
N1 IN THE USER
PROGRAM.
THE POLYNOMIAL IS
SCALED TO AVOID
ARITHMETIC OVERFLOW.
ALL
SCALING USES FACTORS
OF 10. TO CHANGE TO
FACTORS OF 2, SET
BASE
= 2. AND CONST =
LN(2) IN SUBROUTINE
THIS SUBROUTINE USES
DOUBLE PRECISION
ARITHMETIC. SINGLE
PRECISION
IS NOT RECOMMENDED.

01
BASE = 10.
CONST = 2.30259



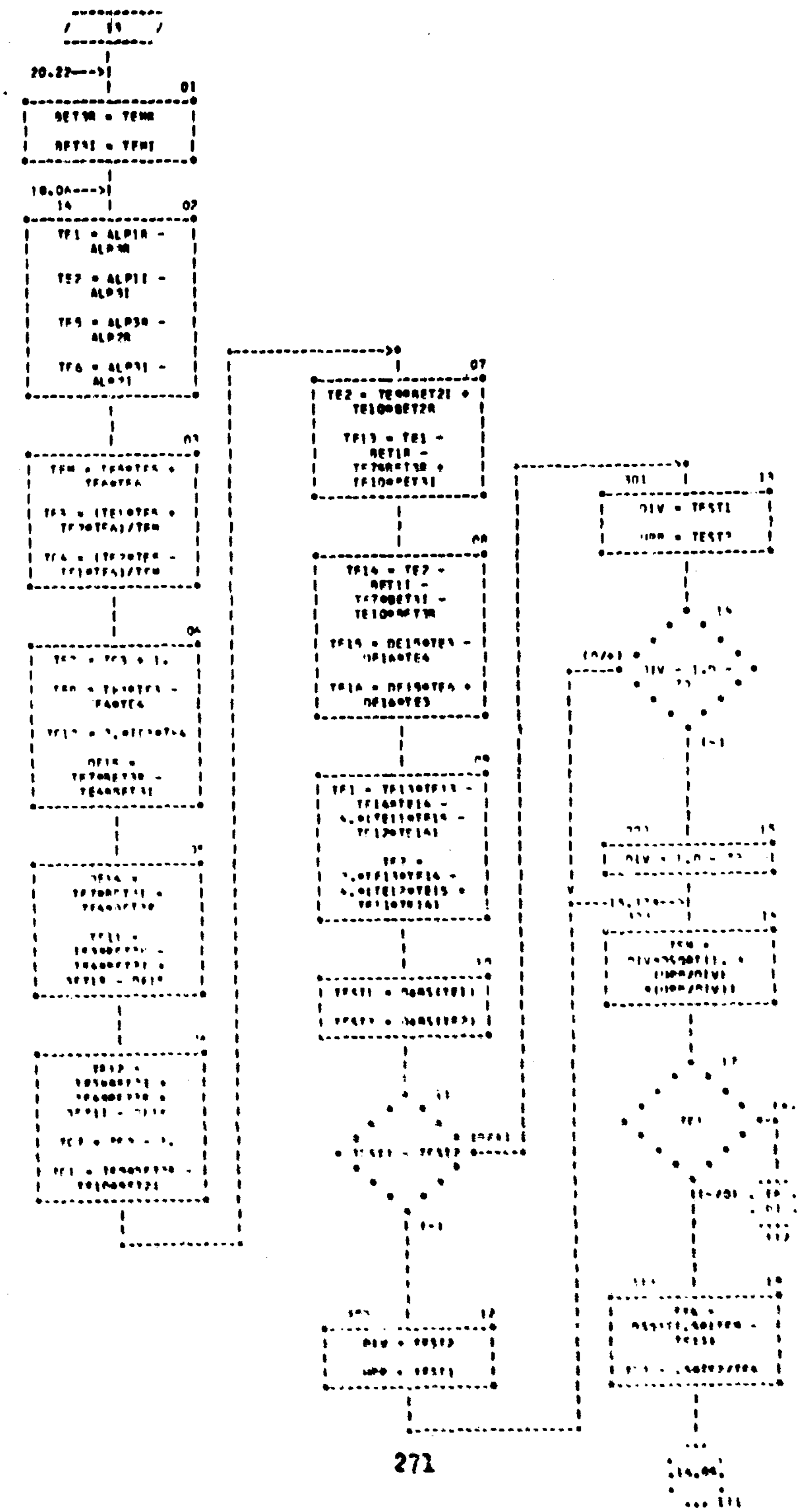
03/11/69 AUTOFLOW CHART SET - FC140 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE MULLER(COF,NI,ROOTR,ROOTI,ISW,IERR)



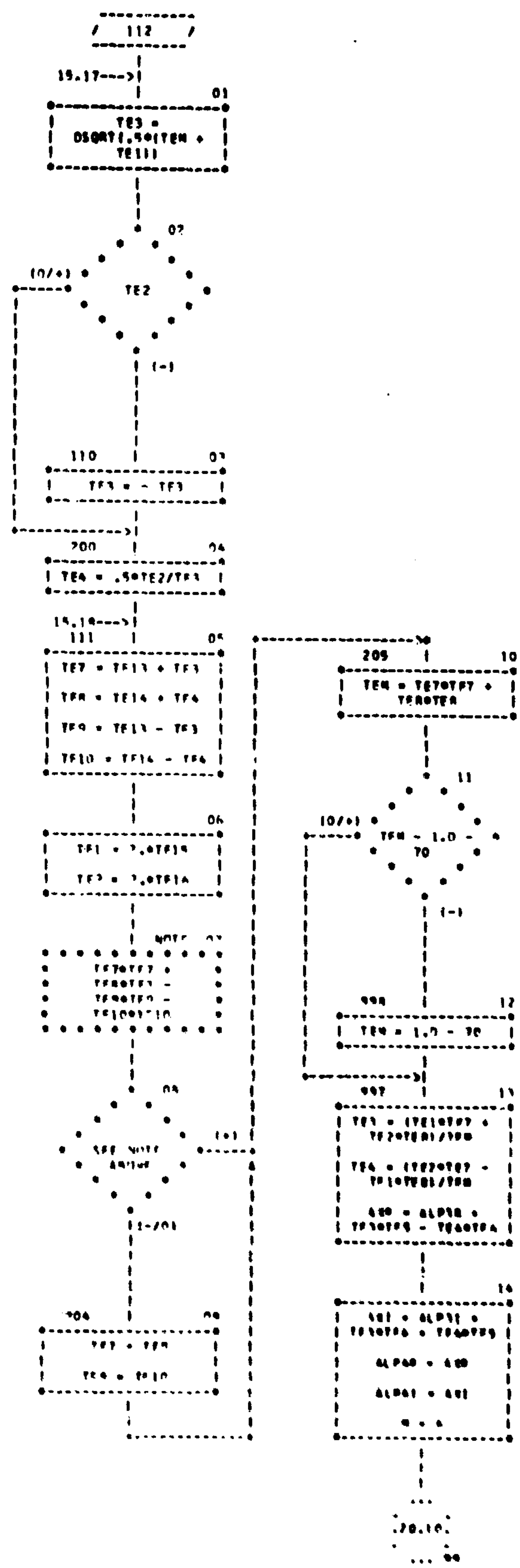
03/11/69 AUTOFLOW CHART SET - EC140 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE MULLER(COF,NI,ROOTR,ROOTI,ISW,IEUR)



03/11/69 AUTOFLOW CHART SET - EC140 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE HULLFR(COE,N1,ROOTR,ROOTI,ISW,IERR)



03/11/69 AUTOFLOW CHART SET - EC140 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE MULLER(COF,N1,ROOTR,ROOTI,ISW,IERR)

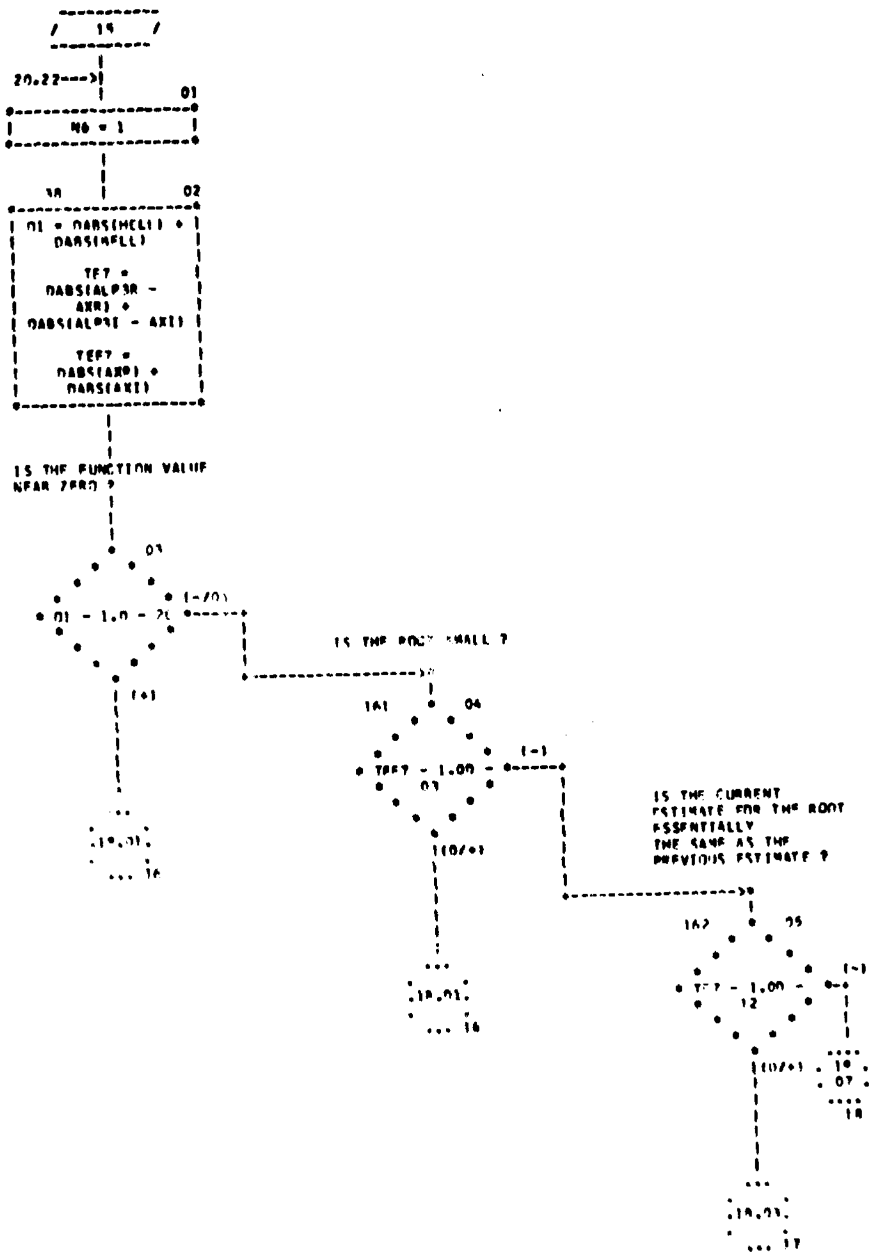
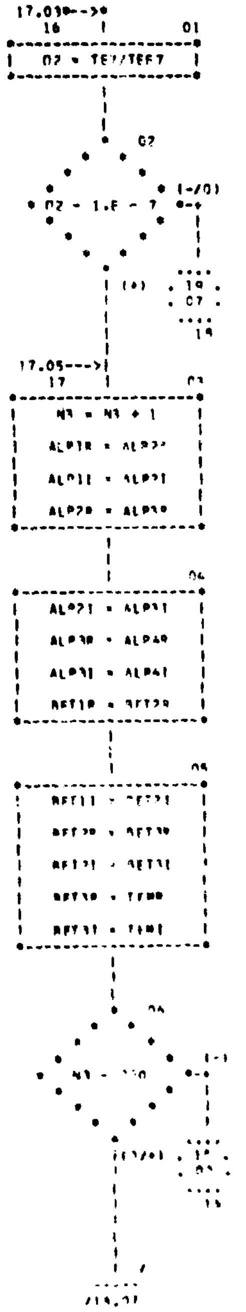
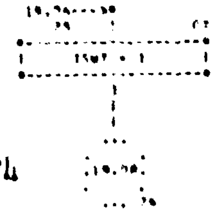


CHART TITLE - SUBROUTINE NULLF(ICOE,N1,ROOTR,ROOTI,ISW,IERR)

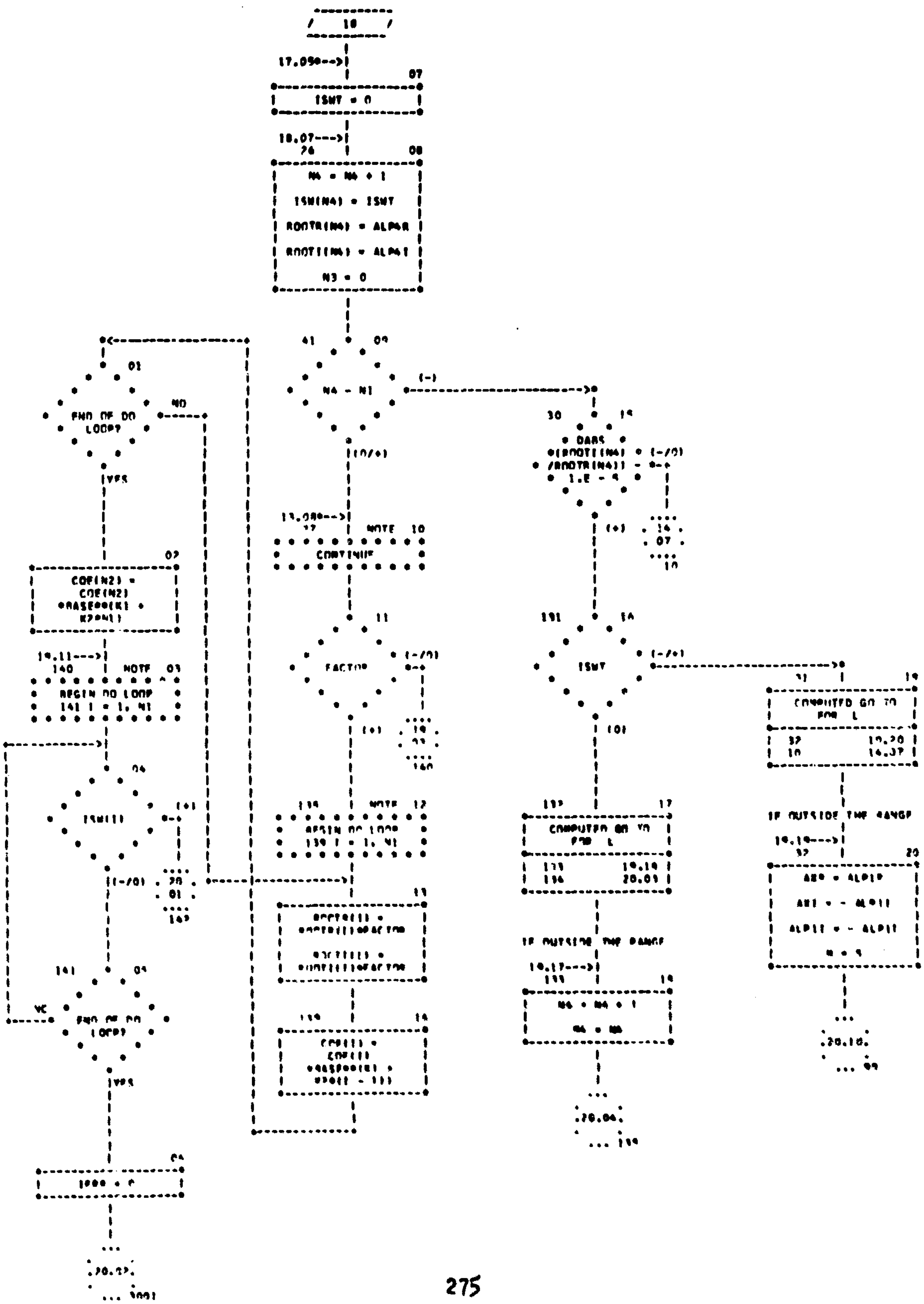
ARE THE CURRENT AND PREVIOUS ESTIMATES OF THE ROOT ESSENTIALLY THE SAME WHEN COMPARED TO THE MAGNITUDE OF THE ROOT ?





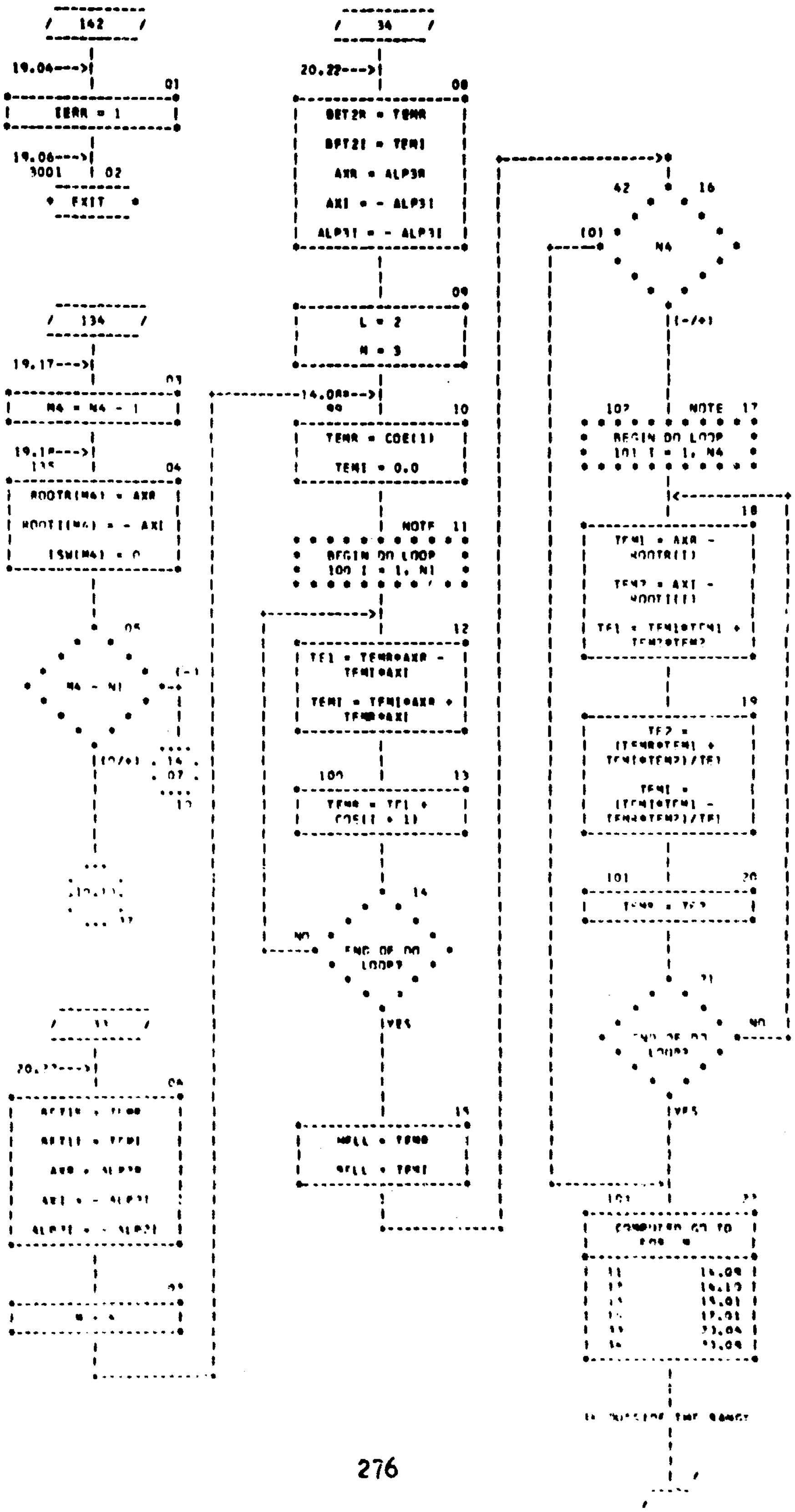
03/11/69 AUTOFLOW CHART SET - EC140 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE MULLER(COE,N1,ROOTR,ROOTI,ISW,IERR)



03/11/69 AUTOFLOW CHART SET - FC140 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE MULLER(COE,NI,ROOTR,ROOTI,ISW,IERR)



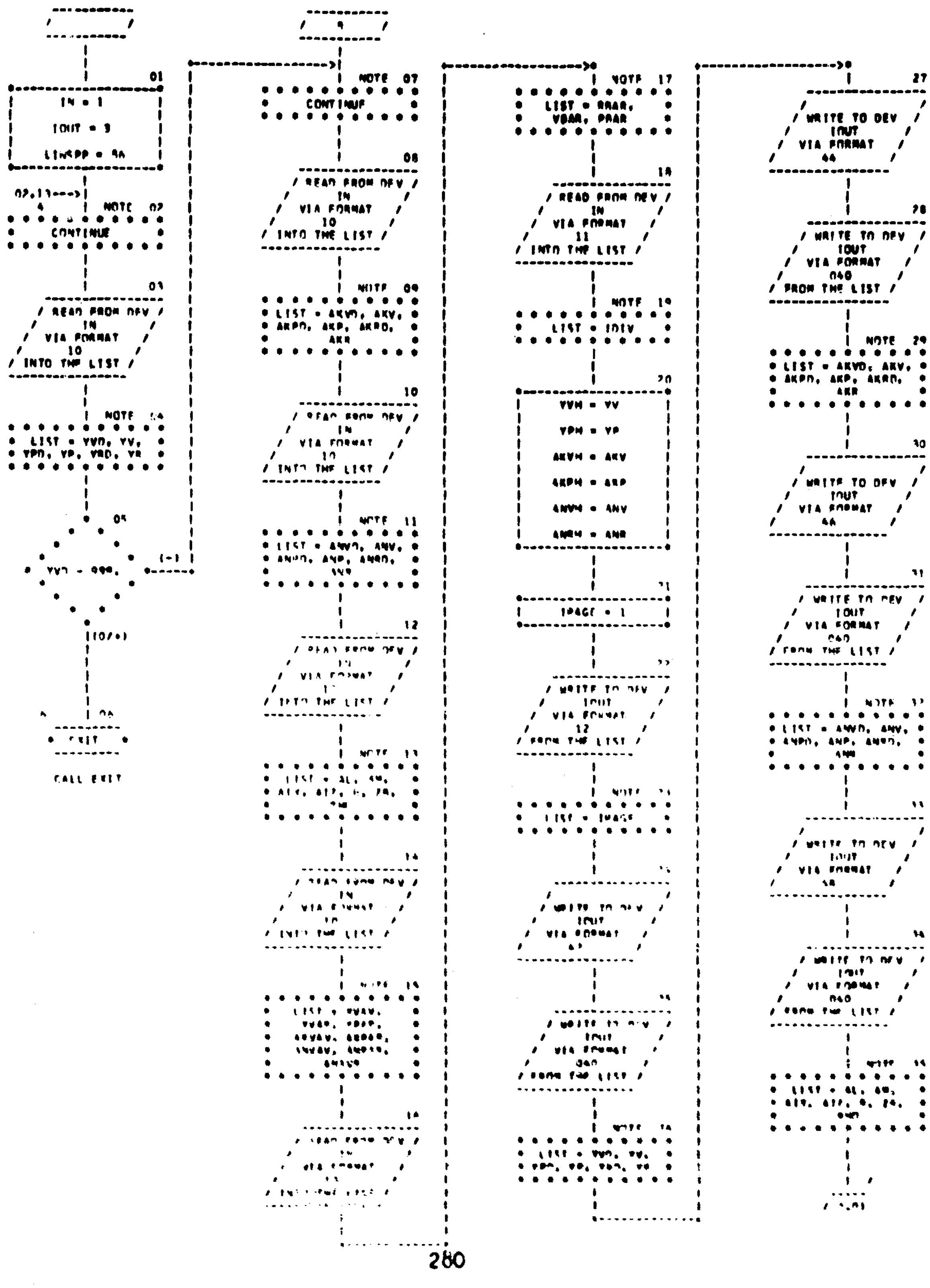
```
//        JOB    EC320
//        EXEC FFORTRAN
      COMMON  A(5,5,25), AA(25), NP11
      DOUBLE PRECISION  A, AA, C(3), A1,A2,A3, ROOTR(10), ROOTI(10)
      DIMENSION ISW(10)
      EQUIVALENCE (C(1),A1),(C(2),A2), (C(3),A3)
      IN = 1
      IOUT = 3
      LINSPP=56
    4 CONTINUE
      READ (IN,10)YVD, YV, YPD, YP, YRD, YR
      IF(YVD-999.)8,6,6
    6 CALL EXIT
    8 CONTINUE
      READ (IN,10)AKVD, AKV, AKPD, AKP, AKRD, AKR
      READ (IN,10)ANVD, ANV, ANPD, ANP, ANRD, ANR
      READ (IN,10) AL, AM,AIX,AIZ, B, ZB, RHO
      READ(IN,10) YVAV, YVAR, YPAP, AKVAV, AKPAP, ANVAV, ANRAR, ANAVR
      READ(IN,10) RBAR, VBAR, PBAR
      READ(IN,11) IDIV
   10 FORMAT(8F10.5)
   11 FORMAT(16I5)
      YVH= YV
      YPH= YP
      AKVH=AKV
      AKPH= AKP
      ANVH= ANV
      ANRH= ANR
      IPAGE = 1
      WRITE(IOUT,12) IPAGE
   12 FORMAT(1H1,50X,'EC320'40X,'PAGE',I10/)
      WRITE(IOUT,42)
      WRITE(IOUT,040)YVD, YV, YPD, YP, YRD, YR
      WRITE(IOUT,44)
      WRITE(IOUT,040)AKVD, AKV, AKPD, AKP, AKPD, AKR
      WRITE(IOUT,46)
      WRITE(IOUT,040)ANVD, ANV, ANPD, ANP, ANRD, ANR
      WRITE(IOUT,48)
      WRITE(IOUT,040)AL,AM, AIX, AIZ,  B, ZB, RHO
      WRITE(IOUT,49)
      WRITE(IOUT,40)YVAV,YVAR,YPAP,AKVAV,AKPAP,ANVAV,ANRAR,ANAVR
      WRITE(IOUT,50)
      WRITE(IOUT,40)RBAR, VBAR, PBAR
   40 FORMAT(1H ,8F16.6)
   42 FORMAT(1H ,8X,'YVD',13X,'YV',14X,'YPD',13X,'YP',14X,'YRD',
     113X,'YR')
   44 FORMAT(1H ,8X,'KVD',13X,'KV',14X,'KPD',13X,'KP',14X,'KRD',
     113X,'KR')
   46 FORMAT(1H ,8X,'NVD',13X,'NV',14X,'NPD',13X,'NP',14X,'NRD',
     113X,'NR')
   48 FORMAT(1H ,8X,'L',15X,'M',15X,'IX',14X,'IZ',14X,'B',15X,'ZB',
     114X,'RHO')
   49 FORMAT(1H ,6X,'YVAV',12X,'YVAR',12X,'YPAP',12X,'KVAV',12X,
     1'KPAP',12X,'NVAV',12X,'NRAR',12X,'NAVR')
```

```
50 FORMAT(1H ,8X,'RBAR',12X,'VBAR',12X,'PBAR')
   WRITE(IOUT,52)
52 FORMAT(1H/)
   WRITE(IOUT,54)
54 FORMAT(1H ,8X,'YV',14X,'YP',14X,'KV',14X,'KP',14X,'NV',14X,'NR'/
  1       1H ,8X,'A1',14X,'A2',14X,'A3',14X,'A4',14X,'A5',14X,'U'/
  21H ,8X,'R',15X,'I',15X,'R',15X,'I',15X,'R',15X,'I',15X.'R',15X,
  3'I'/)
   LINS = 19
56 CONTINUE
   READ (IN,10) U
   IF(U)58,4,58
58 CONTINUE
   U= U* 1.689
   IF(IDIV)62,64,62
62 CONTINUE
   DIV = U
   GO TO 65
64 DIV = 1.
65 CONTINUE
   YV  = YVH +(YVAV*VBAR+AL*YVAR*RBAR)/DIV
   YP  = YPH + AL*YPAP*PBAR/DIV
   AKV=  AKVH+ AKVAV*VBAR/DIV
   AKP=  AKPH +AL*AKPAP*PBAR/DIV
   ANV=  ANVH + ANVAV*VBAR/DIV
   ANR=  ANRH +(AL*ANRAR*RBAR+ANAVR*VBAR)/DIV
   N = 2
   NCOL = 3
   M = N*NCOL+1
   DO 60  I=1,NCOL
   DO 60 J=1,NCOL
   DO 60 K=1,M
60 A(I,J,K)=0.
   RL  = RHO*AL/2
   RL2 = RL* AL
   RL3 = RL2 * AL
   RL4 = RL3 * AL
   RL5 = RL4 * AL
   RL4U = RL4 * U
   RL3U = RL3 * U
   A1  = 0.
   A2  = RL3*YVD-AM
   A3  = RL2*U*YV
   CALL PLACE(1,1,C)
   A2= RL4*YPD
   A3= RL3U*YP
   CALL PLACE(1,2,C)
   A2= RL4*YPD
   A3= RL3U*YR -AM*U
   CALL PLACE(1,3,C)
   A2 = RL4*ANVD
   A3=RL3U*ANV
   CALL PLACE(3,1,C)
   A2=RL5*ANPD
```

```
      A3=RL4U*ANP
      CALL PLACE(3,2,C)
      A2= RL5*ANRD-AIZ
      A3= RL4U*ANR
      CALL PLACE(3,3,C)
      A1 = RL4*AKVD
      A2 = RL3U*AKV
      A3 = 0.
      CALL PLACE(2,1,C)
      A1=RL5*AKPD-AIX
      A2 = RL4U*AKP
      A3 = B*ZB
      CALL PLACE(2,2,C)
      A1=RL5*AKRD
      A2=RL4U*AKR
      A3=0.
      CALL PLACE(2,3,C)
      CALL CHREQN
      NP1=NP11+1
      IF(LINS-LINSPP)80,70,70
   70 IPAGE=IPAGE+1
      WRITE(IOUT,12) IPAGE
      WRITE(IOUT,54)
      LINS=5
   80 WRITE(IOUT,40) YV, YP, AKV, AKP, ANV, ANR
      WRITE(IOUT,40)(AA(I),I=1,NP1), U
      CALL MULLER(AA, NP11,ROOTR,ROOTI,ISW, IERR)
      WRITE(IOUT,40)(ROOTR(I),ROOTI(I),I=1,NP11)
      WRITE(IOUT,52)
      LINS = LINS + 4
      GO TO 56
      END
```

03/11/69 AUTOFLOW CHART SET - EC320 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES



03/11/69

AUTOFLOW CHART SET - EC320 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES

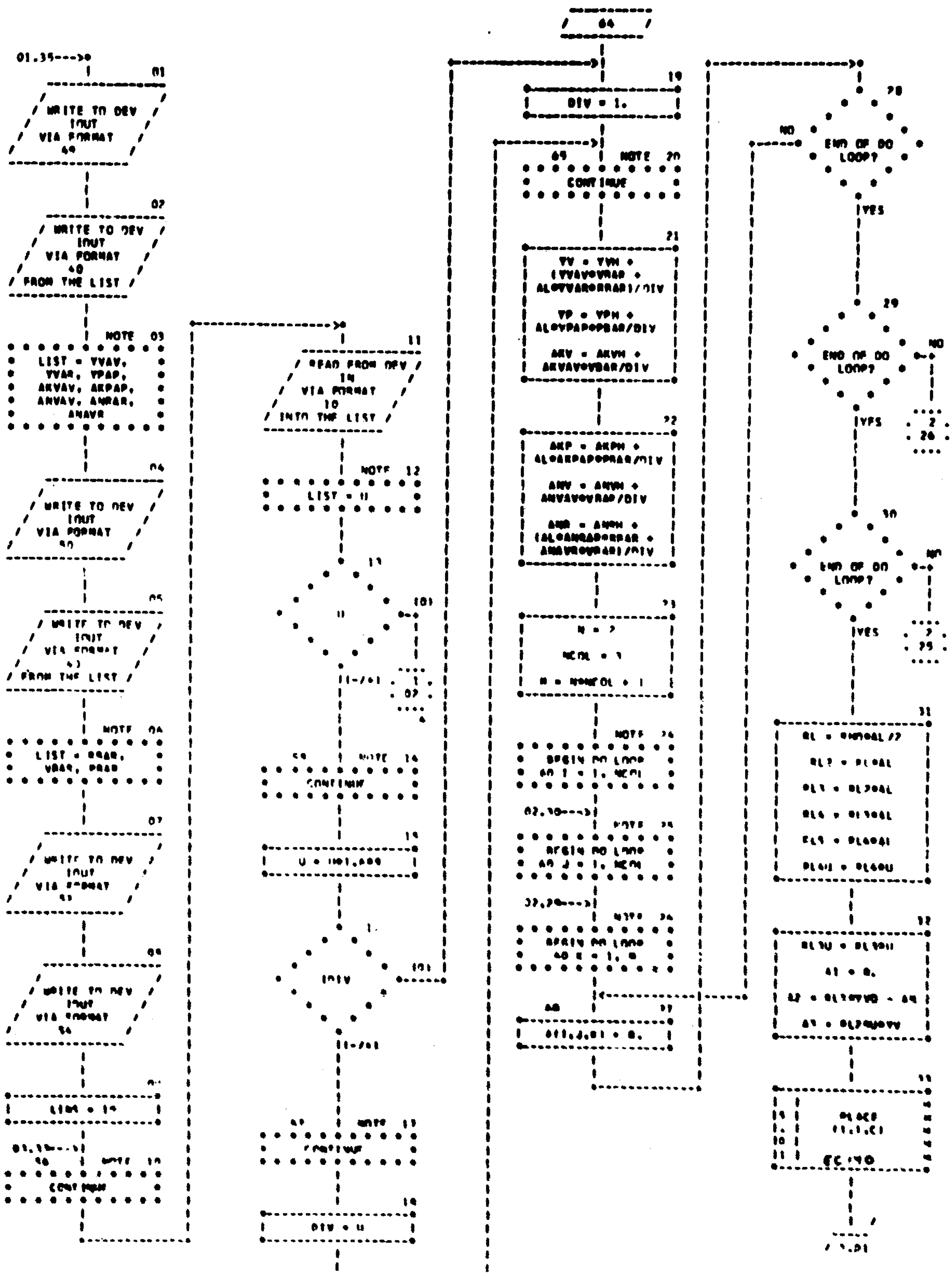
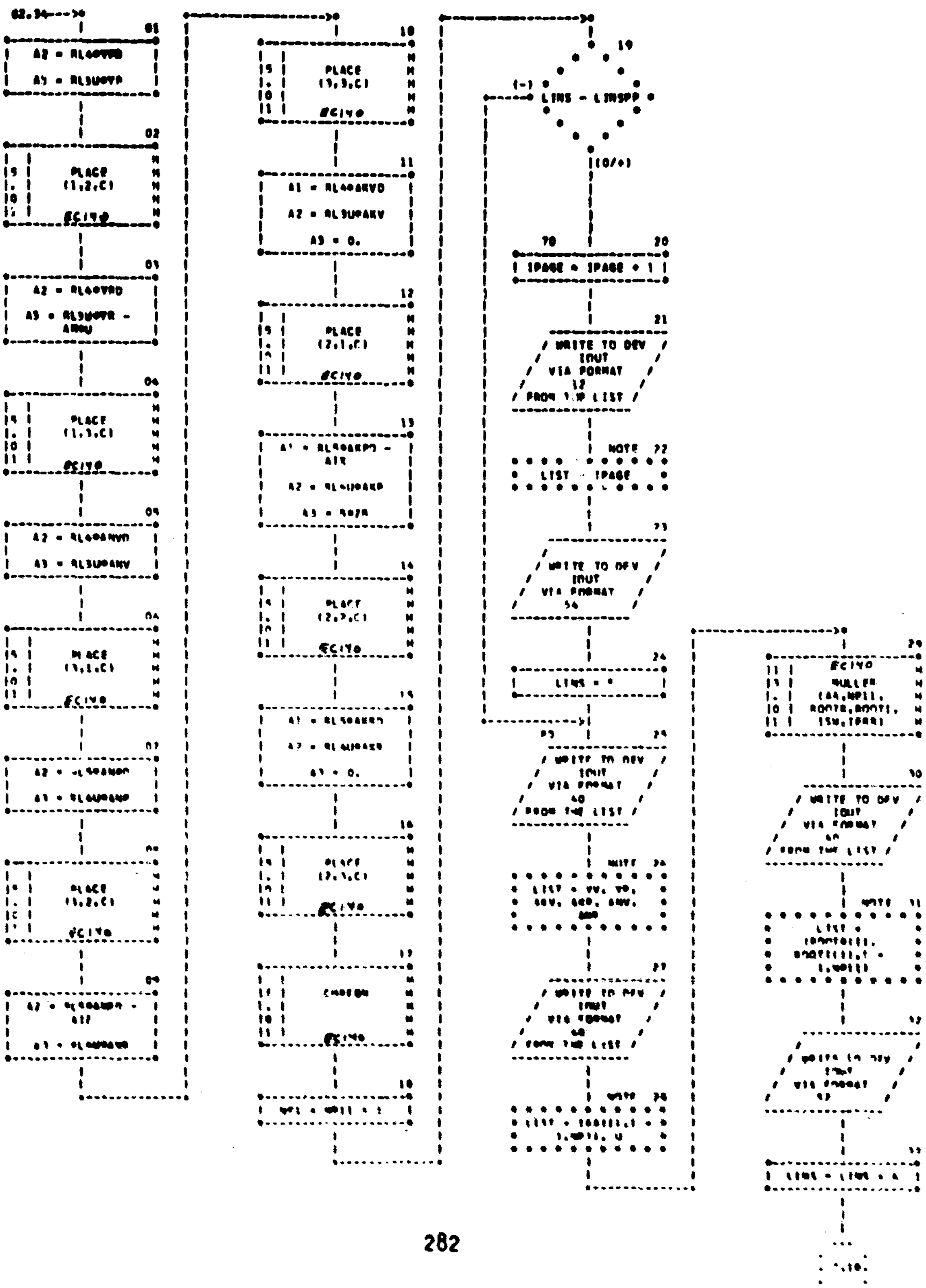


CHART TITLE - PROCEDURES



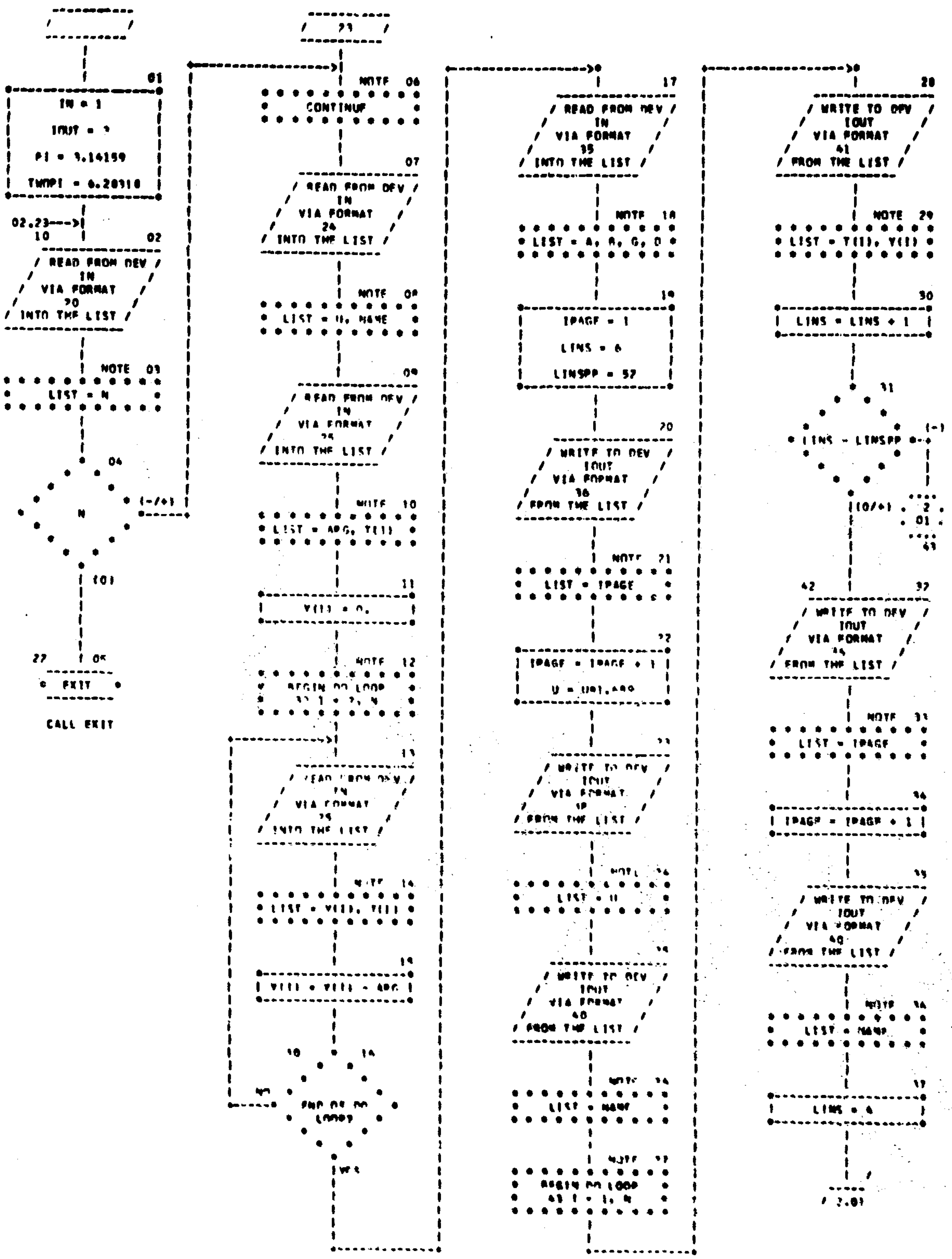
```
//        JOB    EC150
//        EXEC FFORTRAN
      DIMENSION Y(500), T(500), NAME(2)
      IN=1
      IOUT=3
      PI = 3.14159
      TWOPI = 6.28318
   10 READ(IN,20) N
   20 FORMAT(16I5)
      IF(N)23,22,23
   22 CALL EXIT
   23 CONTINUE
   24 FORMAT(F10.5,2A4)
      READ(IN,24) U, NAME
   25 FORMAT(2E15.7)
      READ(IN,25) ARG, T(1)
      Y(1) =0.
      DO 30 I=2,N
      READ(IN,25) Y(I), T(I)
      Y(I) = Y(I)-ARG
   30 CONTINUE
      READ(IN,35) A,B,G,D
   35 FORMAT(8F10.5)
      IPAGE = 1
      LINS  = 6
      LINSPP = 52
      WRITE(IOUT,36) IPAGE
      IPAGE =IPAGE+1
      U = U*1.689
      WRITE(IOUT,38) U
      WRITE(IOUT,40) NAME
   36 FORMAT(1H1,50X,'EC150', 40X,'PAGE',I7/)
   38 FORMAT(1H ,'U= ',E13.6/)
   40 FORMAT(1H ,9X,'T', 14X,2A4/)
   41 FORMAT(1H ,8E16.6)
      DO 43 I=1, N
      WRITE(IOUT,41)  T(I), Y(I)
      LINS = LINS + 1
      IF(LINS-LINSPP)43,42,42
   42 WRITE(IOUT,36) IPAGE
      IPAGE = IPAGE + 1
      WRITE(IOUT,40) NAME
      LINS = 4
   43 CONTINUE
C  FIND LAST MIN
      DO 100 I=1,N
      J = N-I+1
      IF( Y(J)- Y(J-1))100,100,50
   50 IF(Y(J-1) - Y(J-2))110,100,100
  100 CONTINUE
  110 YMIN = Y(J-1)
      T5 = T(J-1)
      K = J-1
C  FIND PRECEEDING MAX
```

```
      DO 150 I = 1,K
      J = K-I+1
      IF(Y(J)-Y(J-1))125,150,150
  125 IF(Y(J-1)-Y(J-2))150,150,160
  150 CONTINUE
  160 YMAX = Y(J-1)
      T4 = T(J-1)
C   COMPUTE  TO
      TZ = 4.61/D
      PEST = AMOD(B*T4,TWOPI)
      PEST = PEST - PI
      TEMP1 = EXP( G*(T5-T4))
      TEMP2 = EXP(-A*T4)
      A2EST =-(YMAX-YMIN*TEMP1)/(TEMP1*EXP(-A*T5)+TEMP2)
      A1EST =(YMAX+A2EST*TEMP2)/EXP(-G*T4)
      A3EST = -(A1EST+A2EST*COS(PEST))
      WRITE(IOUT,36) IPAGE
      IPAGE = IPAGE + 1
      WRITE(IOUT,200)YMIN,YMAX,TZ,T4,T5
      WRITE(IOUT,204) A,B,G,PEST,A1EST,A2EST,D, A3EST
  200 FORMAT(1H ,9X,'YMIN',11X,'YMAX',11X,'TO',13X,'T4',13X,'T5'/
     1      1H ,5E16.6/)
  204 FORMAT(1H ,9X,'A',14X,'B',14X,'G',14X,'P',14X,'A1',13X,'A2',
     113X,'D',14X,'A3'/1H ,8E16.6/)
      GO TO 10
      END
/*
/&
```

03/11/69

AUTOFLOW CHART SET - EC150 NAVTRADEVCEN 68-C-0050-2

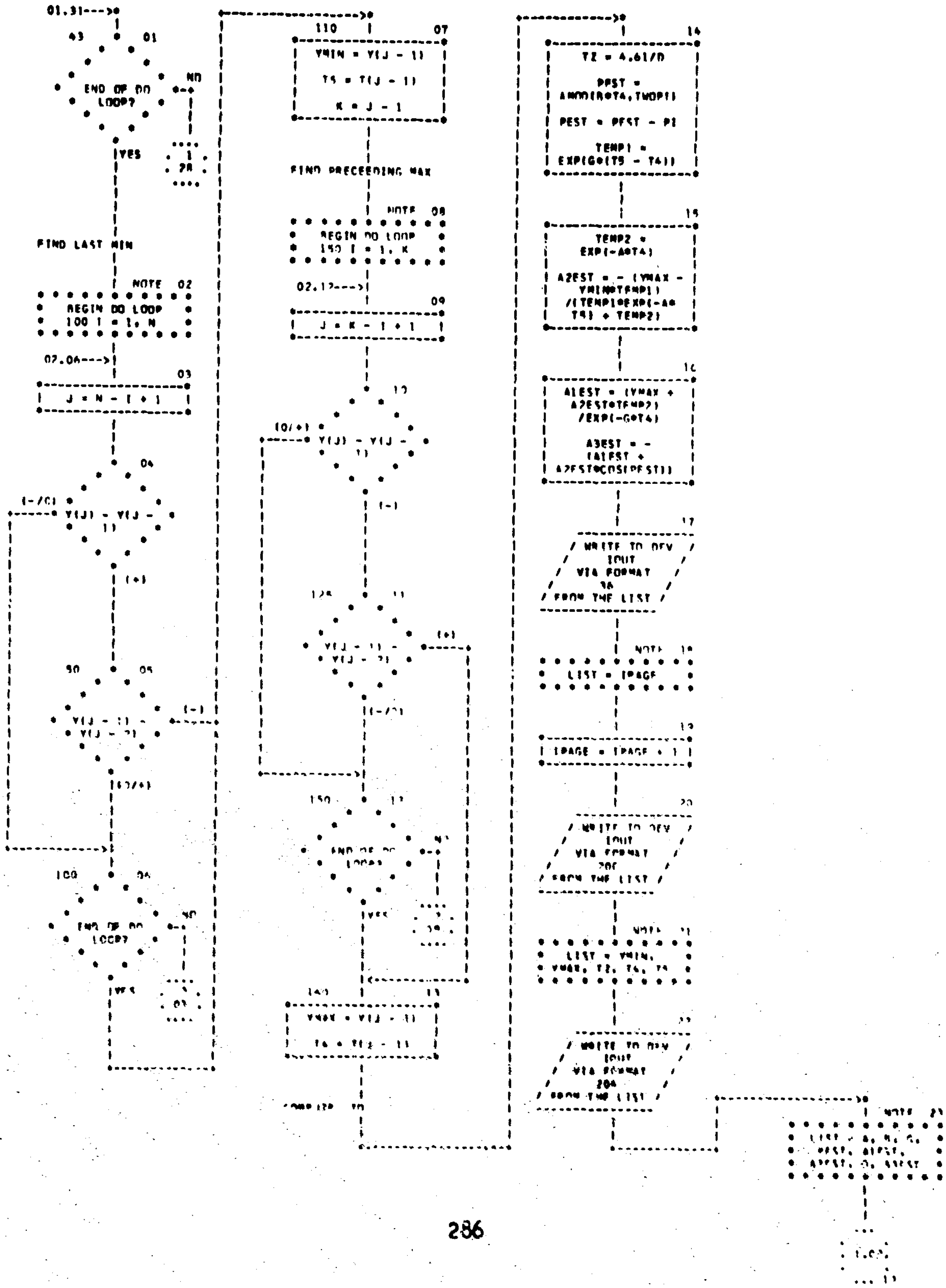
CHART TITLE - PROCEDURES



03/11/69

AUTOFLOW CHART SET - FL150 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES



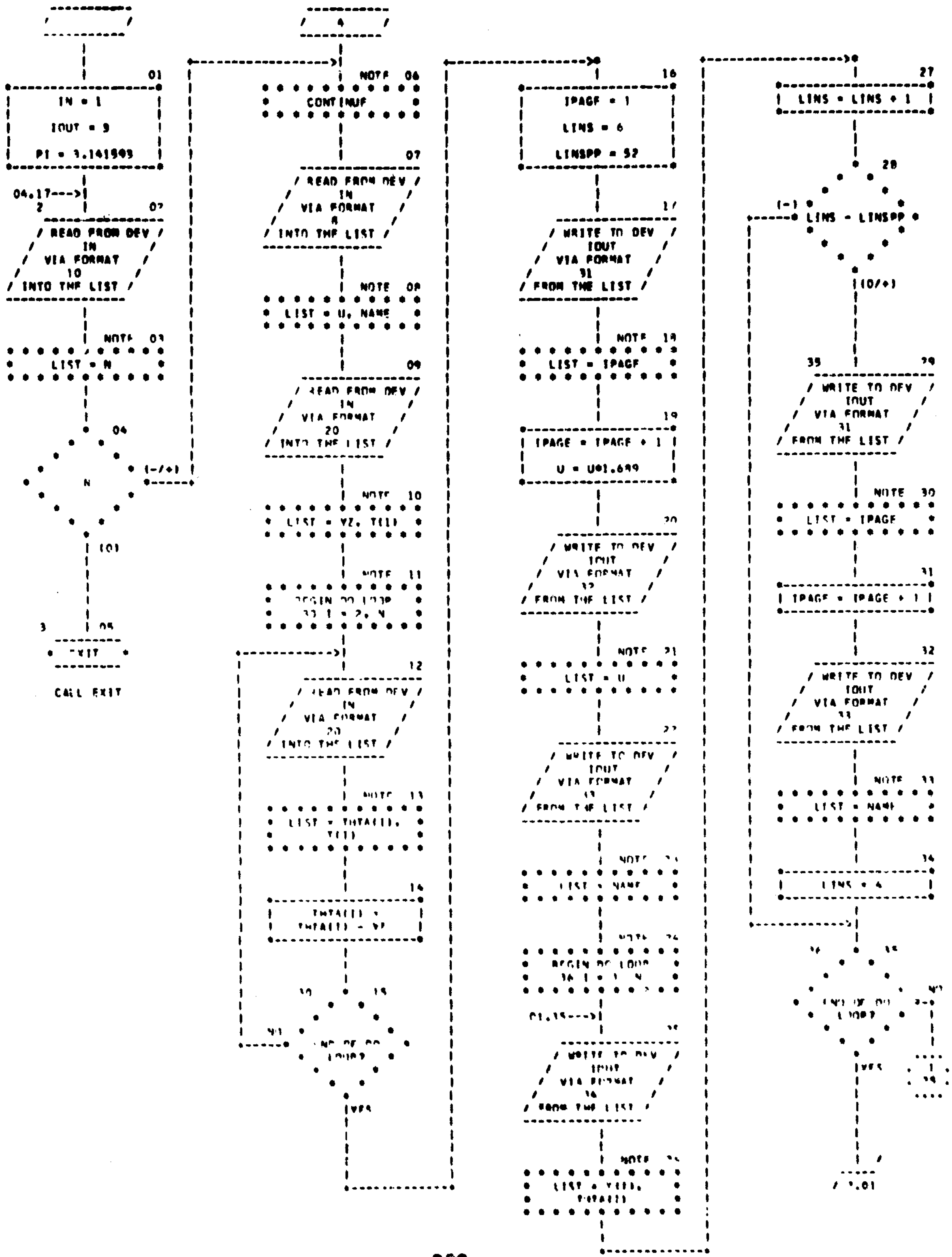
```
//       JOB    EC330
//       EXEC FFORTRAN
      DIMENSION THTA(500), T(500), NAME(2)
      IN= 1
      IOUT = 3
      PI= 3.141593
    2 READ(IN,10) N
      IF(N)4,3,4
    3 CALL EXIT
    4 CONTINUE
      READ(IN,8) U, NAME
    8 FORMAT(F10.5,2A4)
   10 FORMAT(16I5)
      READ(IN,20) YZ, T(1)
   20 FORMAT(2E15.7)
      DO 30 I=2,N
      READ(IN,20) THTA(I), T(I)
      THTA(I)= THTA(I)- YZ
   30 CONTINUE
      IPAGE=1
      LINS=6
      LINSPP=52
      WRITE(IOUT,31)IPAGE
   31 FORMAT(1H1,50X,'EC330',40X,'PAGE',I7/)
      IPAGE = IPAGE + 1
      U = U*1.689
      WRITE(IOUT,32) U
   32 FORMAT(1H ,'U= ',F13.6/)
      WRITE(IOUT,33) NAME
   33 FORMAT(1H ,9X,'T',14X,2A4/)
      DO 36 I=1,N
      WRITE(IOUT,34)T(I), THTA(I)
   34 FORMAT(1H ,8E16.6)
      LINS = LINS + 1
      IF(LINS-LINSPP)36,35,35
   35 WRITE(IOUT,31)IPAGE
      IPAGE = IPAGE + 1
      WRITE(IOUT,33)NAME
      LINS = 4
   36 CONTINUE
      WRITE(IOUT,31) IPAGE
C   FIND TIME OF FIRST CROSSING
      DO 40 I=6, N
      IF (THTA(I)) 50, 40,40
   40 CONTINUE
   50 Y1 = THTA(I-1)
      Y2 = THTA(I)
      X1 = T(I-1)
      X2 = T(I)
      T1 = X1+ Y1*(X2-X1)/(Y1-Y2)
C   FIND TIME OF SECOND CROSSING
      DO 60 K= I, N
      IF(THTA(K))60,60,70
   60 CONTINUE
```

```
   70 Y1  =  THTA(K-1)
      Y2  =  THTA(K)
      X1  =  T(K-1)
      X2  =  T(K)
      T2  =  X1+ Y1*(X2-X1)/(Y1-Y2)
C  COMPUTE  PERIOD
      P =   T2-T1
C  COMPUTE  BETA
      BETA = PI/P
C  COMPUTE  PSI
      PSI  = BETA*T1- PI/2.
C  COMPUTE  T3 AND T4
      TEMP  = P/2.
      T3  = T1-TEMP
      T4  = T1+TEMP
C  FIND  Y3
      DO 80 I = 1,N
      IF(T(I)-T3)80,80,90
   80 CONTINUE
   90 Y1  =  THTA(I-1)
      Y2  =  THTA(I)
      X1  =  T(I-1)
      X2  =  T(I)
      Y3  =  Y1+(T3-X1)*(Y2-Y1)/(X2-X1)
C  FIND Y4
      DO 100 K= I,N
      IF(T(K)-T4)100,100,110
  100 CONTINUE
  110 Y1  =  THTA(K-1)
      Y2  =  THTA(K)
      X1  =  T(K-1)
      X2  =  T(K)
      Y4  =  Y1  +(T4-X1)*(Y2-Y1)/(X2-X1)
C  COMPUTE  ALPHA
      ALPHA  =  ALOG(-Y3/Y4)/P
C  COMPUTE  A2
      A2  =   Y3/EXP(-ALPHA*T3)
C  COMPUTE  A1
      A1  =   -A2* COS(PSI)
C  COMPUTE  GAMA
      TEMP  = A2*EXP(-ALPHA*6.)*COS(BETA*6.-PSI)
      ARG =(THTA(4)-TEMP)/A1
      IF(ARG)112,112,114
  112 GAMAN  =  10.* ALPHA
      ALPN   =  ALPHA
      WRITE(IOUT,170)  ALPHA,BETA,GAMAN,PSI,A1,A2
      GO TO 126
  114 CONTINUE
      GAMA  =  -ALOG(ARG)/6.
      WRITE(IOUT,170)ALPHA,BETA,GAMA,PSI,A1,A2
      TEMP1=  A1*EXP(-GAMA*T3)
      TEMP2  =  Y3-TEMP1
      ALPN  =      TEMP2  /(-Y4+A1*EXP(-GAMA*T4))
      ALPN  =  ALOG(ALPN)/P
```

```
      A2N  = TEMP2/EXP(-ALPN*T3)
      A1N  = -COS(PSI)* A2N
      TEMP =  A2N*EXP(-ALPN*6.)*COS(BETA*6.-PSI)
      ARG = (THTA(4)-TEMP)/A1N
      IF(ARG) 121,121,122
  121 GAMAN = 10. * ALPN
      GO TO 125
  122 GAMAN = - ALOG(ARG) /6.
  125 CONTINUE
      WRITE(IOUT,170)ALPN,BETA,GAMAN,PSI,A1N,A2N
  126 CONTINUE
C  FIND  MAX  ORD
      DO 130 J= 2,N
      IF(THTA(J)-THTA(J+1))130,130,140
  130 CONTINUE
  140 T3P = T(J)
      Y3P = THTA(J)
C  FIND  MIN  ORD
      DO 150 K= J,N
      IF(THTA(K))145,150,150
  145 IF(THTA(K)-THTA(K+1))160,150,150
  150 CONTINUE
  160 T4P = T(K)
      Y4P = THTA(K)
      A2 =Y3P/(-COS(PSI)*EXP(-GAMAN*T3P)+EXP(-ALPN*T3P)*
     1COS(BETA*T3P-PSI))
      A1 = -A2*COS(PSI)
      WRITE(IOUT,170) ALPN,BETA,GAMAN,PSI,A1,A2
  170 FORMAT(//1H ,9X,'A',15X,'B',15X,'G',15X,'P',15X,'A1',14X,'A2'//
     11H ,6F16.6//)
      GO TO 2
      END
/*
/&
```

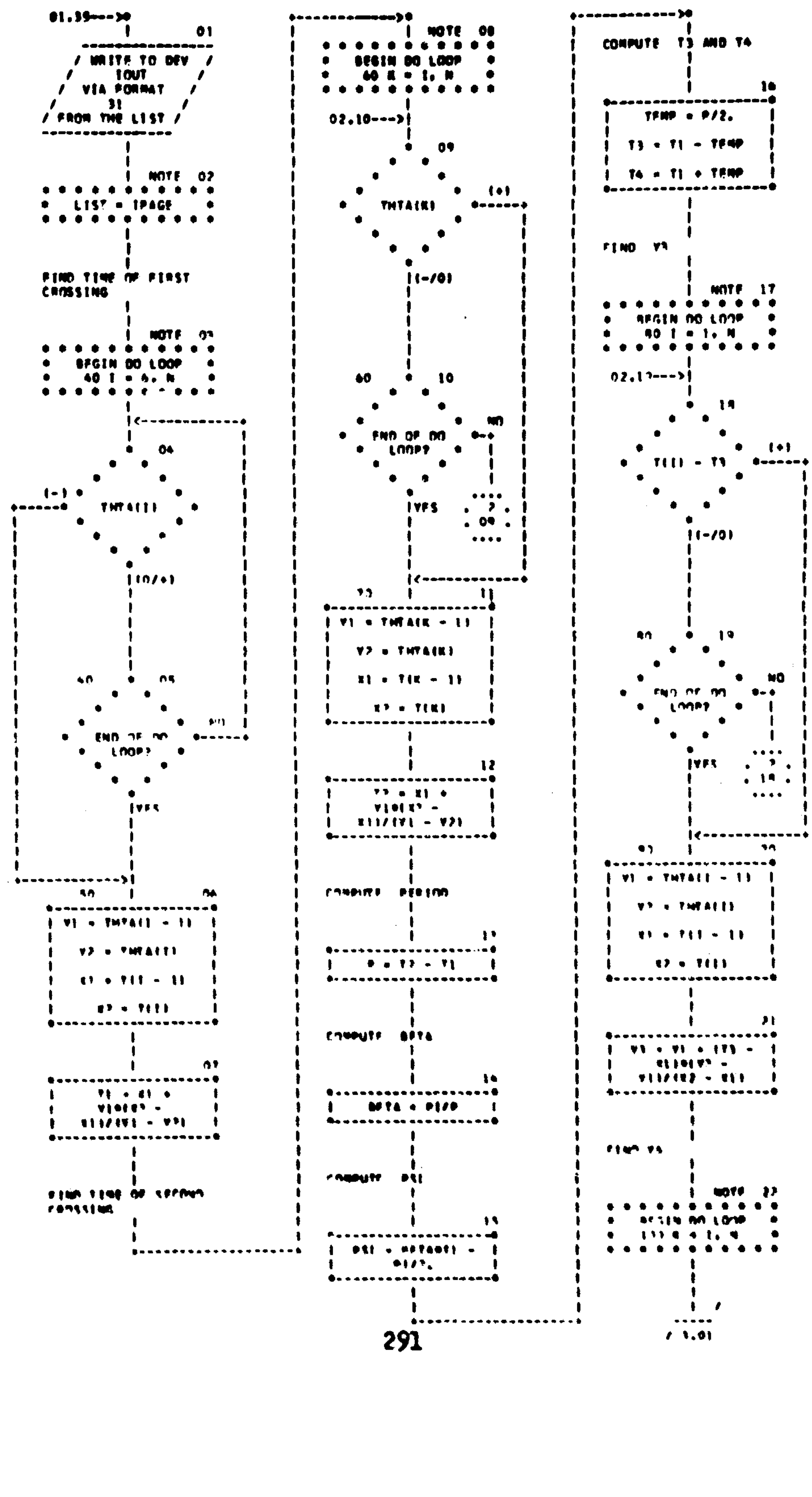
03/11/69 AUTOFLOW CHART SET - EC330 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES



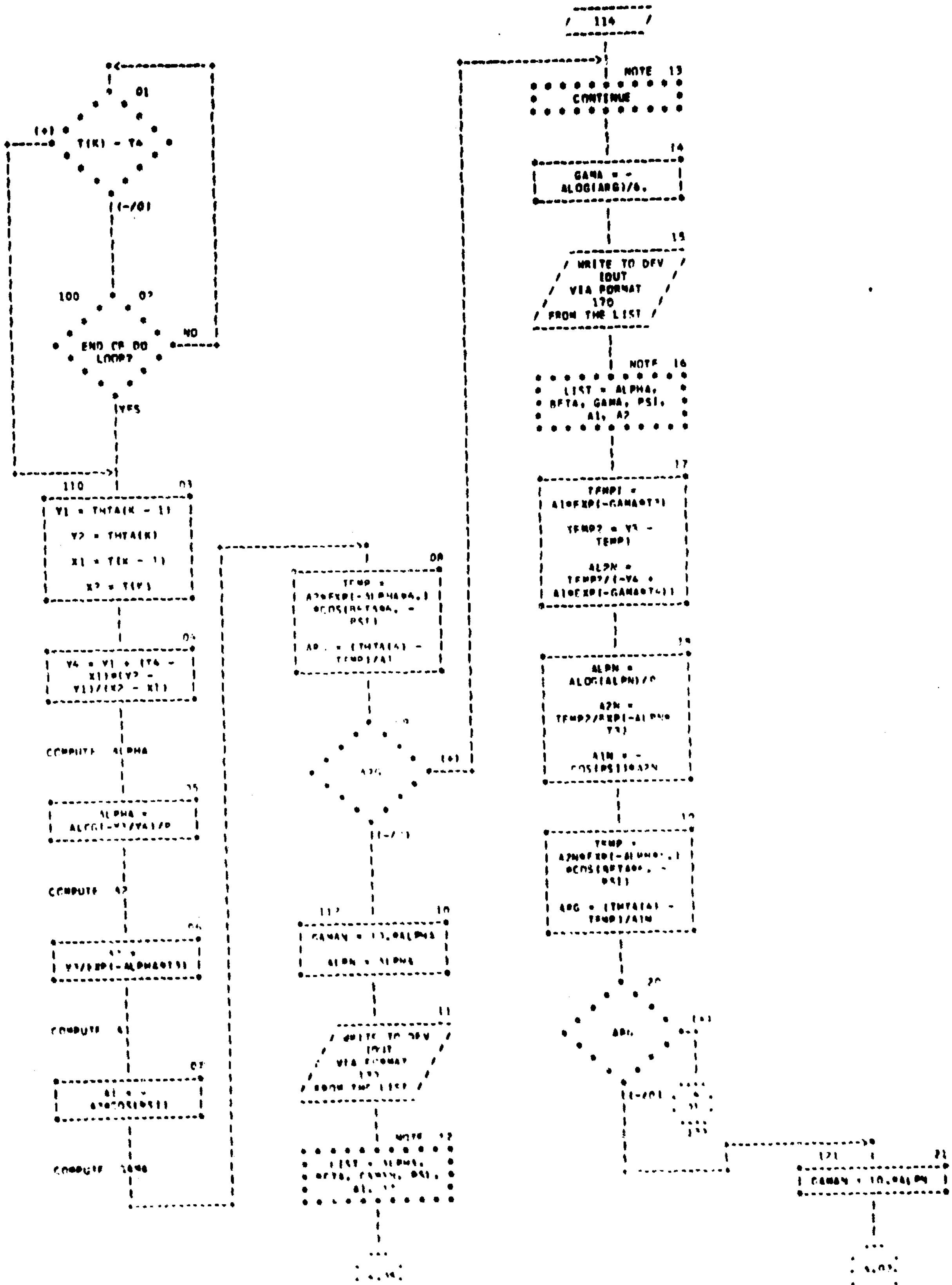
03/11/69 AUTOFLOW CHART SET - FC330 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES



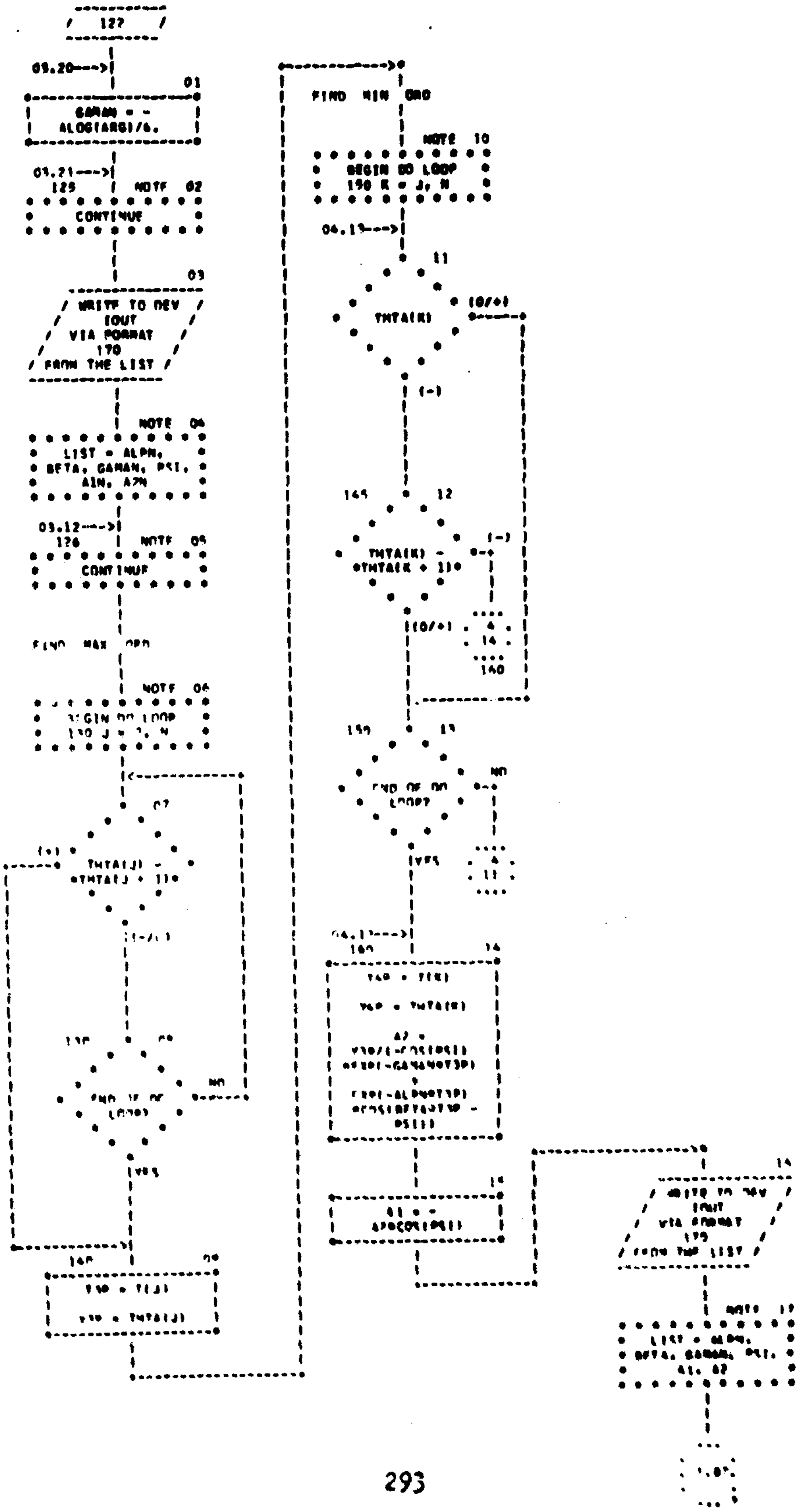
09/11/69 AUTOFLOW CHART SET - EC930 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES



03/11/69 AUTOFLOW CHART SET - EC330 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES



```
//        JOB   EC310
//        EXEC FFORTRAN
      DIMENSION  T(500), Y(500),  X(10)
      DIMENSION BUFF(3000), FCN1(500), IY(2)
      COMMON T,Y,NPTS,FCN1,ISW1
      IN = 1
      IOUT = 3
      IOPEN  =  0
    3 CONTINUE
      LINS = 99
      LINSPP = 52
      READ(IN,5) N, NUMSIG, MAXIT, IPRINT, NPTS, ISW1, IPLOT
    5 FORMAT(16I5)
      IF(IPLOT)6,8,6
    6 IF(IOPEN) 8,7,8
    7 CALL PLOTS(BUFF, 12000, 7)
      IOPEN = 1
    8 CONTINUE
      IF(N)20,10,20
   10 CONTINUE
      IF(IOPEN)12,15,12
   12 CONTINUE
      CALL PLOT(6.0,0.0,999)
   15 CONTINUE
      CALL EXIT
   20 CONTINUE
      READ(IN,22) U, IY
   22 FORMAT(F10.5,2A4)
      U = U*1.689
      READ(IN,25) ARG, T(1)
      Y (1) = 0.
      DO 30 I = 2,NPTS
      READ(IN,25 ) Y(I),T(I)
      Y(I) = Y(I) - ARG
   25 FORMAT(2E15.7)
   30 CONTINUE
C    READ INITIAL  A, B, G, P, A1, A2, D, A3
      READ(IN,35 ) (X(I), I = 1,8)
   35 FORMAT(8F10.5)
      IF(ISW1)40,32,40
   32 CONTINUE
      CALL SYSTEM(N, NUMSIG, MAXIT, IPRINT, X)
      ISW1=1
   40 CONTINUE
      CALL AUXFCN(X,DUM,IDUM)
      DO 80 I = 1, NPTS
      IF(LINSPP - LINS)42,42,46
   42 LINS = 0
      WRITE(IOUT, 44)(X(K),K=1,8)
   44 FORMAT(1H1,9X,'A',15X,'B',15X,'G',15X 'P',15X,'A1',14X,'A2',14X,
     1 'D',15X,'A3'/1H ,8E16.6,//1H ,14X,'T',19X,'YE',19X,'YT'/)
   46 CONTINUE
      WRITE(IOUT, 48) T(I), FCN1(I), Y(I)
      LINS = LINS + 1
```

```
48 FORMAT(1H ,3E20.6)
80 CONTINUE
   IF(IPLOT) 50, 3, 50
50 CONTINUE
   CALL FCNPLT(T,FCN1,Y,NPTS,X,U,8, IY)
   WRITE(IOUT,90)
90 FORMAT(1H1)
   GO TO 3
   END
```

```
      SUBROUTINE FCNPLT(X, YF, YT, N, VAR, U, NVAR, IY)
      DIMENSION  VAR(1)
      DIMENSION  X(1), YF(1), YT(1), DUMX(2), DUMY(2)
      DIMENSION ITAB(3), IFIT(2), IWK(10), IY(1)
      IU =IHEX(14,4,4,0,7,14,4,0)
      IWK(1)=IHEX(12,1,4,0,7,14,4,0)
      IWK(2)=IHEX(12,2,4,0,7,14,4,0)
      IWK(3)=IHEX(12,7,4,0,7,14,4,0)
      IWK(4)=IHEX(13,7,4,0,7,14,4,0)
      IWK(5)=IHEX(12,1,15,1,7,14,4,0)
      IWK(6)=IHEX(12,1,15,2,7,14,4,0)
      IWK(7)=IHEX(12,4,4,0,7,14,4,0)
      IWK(8)=IHEX(12,1,15,3,7,14,4,0)
      IDSH=IHEX(6,0,6,0,6,0,4,0)
      ITAB(1)=IHEX(14,3,12,1,12,2,14,4)
      ITAB(2)=IHEX(13,3,12,1,14,3,12,5)
      ITAB(3)=IHEX(12,4,4,0,4,0,4,0)
      IFIT(1)=IHEX(12,6,12,9,14,3,14,3)
      IFIT(2)=IHEX(12,5,12,4,4,0,4,0)
      IX=IHEX(14,3,4,0,4,0,4,0)
      DIV=20.
      DUMX(1)=X(1)
      DUMX(2) = X(N)
      YMAX=-999.
      YMIN=999.
      DO 100  I=1,N
      IF(YF(I)-YMAX)40,40,30
   30 YMAX=YF(I)
   40 IF(YF(I)-YMIN)50,60,60
   50 YMIN=YF(I)
   60 IF(YT(I)-YMAX)80,80,70
   70 YMAX=YT(I)
   80 IF(YT(I)-YMIN)90,100,100
   90 YMIN=YT(I)
  100 CONTINUE
      DUMY(1)=YMIN
      DUMY(2)=YMAX
      CALL PLOT(0.0,1.0,23)
      CALL SCALE(DUMX,8.5,2,1,DIV,2)
      CALL SCALE(DUMY,6.0,2,1,DIV,1)
      CALL AXIS(6.0,0.0,IX,-4,8.5,90.0,DIV,2)
      CALL AXIS(6.0,0.0,IY,8,6.0,180.,DIV,1)
      HIGH = .134
      STR= .5
      XH = 6.5
      CALL SYMBOL(STR,XH,HIGH,IU,90.0,4)
      CALL NUMBER(-0.0,-0.0,-0.0,U,90.,3)
      YH = STR + .25
      DO 110 I = 1,NVAR
      IF(VAR(I)) 105,111,105
  105 CONTINUE
      CALL SYMBOL(YH,XH, HIGH,IWK(I),90.0, 4)
      CALL NUMBER(-0.0, -0.0, -0.0, VAR(I),90.0, 6)
      YH = YH + .25
```

```
110 CONTINUE
111 CONTINUE
    CALL SYMBOL(YH,XH, HIGH, IDSH, 90.0, 4)
    CALL SYMBOL(-0.0,-0.0,-0.0, ITAB, 90.0, 12)
    YH = YH + .125
    CALL PLOT(YH,XH, 3)
    CALL PLOT(YH,XH+.3125,2)
    YH= YH+.125
    CALL SYMBOL(YH, XH+.5, HIGH, IFIT, 90.0,8)
    DO 120 I = 1,N
    YT(I) = -YT(I)
    YF(I) = -YF(I)
120 CONTINUE
    CALL LINE(RMIN,DL,0,-1,1,0)
    RMIN = -(RMIN+DL*6.0)
    CALL LINE(RMIN,DL,0,1,1,0)
    CALL LINE(YT,X, N, 1, 0,050500)
    CALL LINE(YF,X, N, 1, 0, 0)
    CALL PLOT(8.5,-1.0,-23)
    RETURN
    END
```

```
      SUBROUTINE AUXFCN( X, F, K)
      DIMENSION T(500), Y(500), X(1)
      COMMON T,Y,NPTS,FCN1(500),ISW1
      A= X(1)
      B= X(2)
      G= X(3)
      P= X(4)
      A1=X(5)
      A2=X(6)
      D = X(7)
      A3= X(8)
      F=0.
      DO 80 I=1,NPTS
      AT=-A*T(I)
      BT=B*T(I)
      GT=-G*T(I)
      DT= -D*T(I)
      EAT=EXP(AT)
      EGT=EXP(GT)
      EDT = EXP(DT)
      T1=BT-P
      CSN=COS(T1)
      SSN=SIN(T1)
      T2=EAT*CSN
      T3=A2*T2
      T4=A1*EGT
      T5=A2*EAT*SSN
      T6 = A3*EDT
      FCN = T4+T3+T6 -Y(I)
      IF(ISW1)2,4,2
    2 FCN1(I)=T4+T3+T6
      GO TO 80
    4 CONTINUE
      GO TO (10,20,30,40,50,60,62,64), K
   10 CONTINUE
      PAR=-T(I)*T3
      GO TO 70
   20 CONTINUE
      PAR=-T(I)*T5
      GO TO 70
   30 CONTINUE
      PAR=-T(I)*T4
      GO TO 70
   40 CONTINUE
      PAR= T5
      GO TO 70
   50 CONTINUE
      PAR=EGT
      GO TO 70
   60 CONTINUE
      PAR=T2
      GO TO 70
   62 CONTINUE
      PAR = -T(I)*T6
```

```
  161 CALL AUXFCN(X,FPLUS,K)
      PART(ITEMP)=(FPLUS-F)/H
      X(ITEMP)=HOLD
      IF(ABS(PART(ITEMP))) 305,310,305
  305 IF(ABS(F/PART(ITEMP)) - 1.0E+20) 200,200,310
  310 ITALLY=ITALLY+1
  200 CONTINUE
      IF(ITALLY - N + K)202,202,311
  311 CONTINUE
      FACTOR=FACTOR*10.0
      IF(FACTOR - .15) 135,135,775
  202 IF(K - N) 203,312,312
  312 CONTINUE
      IF(ABS(PART(ITEMP)))313,775,313
  313 CONTINUE
      COE(K,N+1)=0.0
      KMAX=ITEMP
      GO TO 500
C
C     FIND PARTIAL DERIVATIVE OF LARGEST ABSOLUTE VALUE.
C
  203 KMAX=LOOKUP(K,K)
      DERMAX=ABS(PART(KMAX))
      KPLUS=K+1
      DO 210 I=KPLUS,N
      JSUB=LOOKUP(K,I)
      TEST=ABS(PART(JSUB))
      IF(TEST-DERMAX) 209,314,314
  314 CONTINUE
      DERMAX=TEST
      LOOKUP(KPLUS,I)=KMAX
      KMAX=JSUB
      GO TO 210
  209 LOOKUP(KPLUS,I)=JSUB
  210 CONTINUE
      IF(ABS(PART(KMAX)))315,775,315
  315 CONTINUE
C
C     SET UP COEFFICIENTS FOR KTH ROW OF TRIANGULAR LINEAR SYSTEM USED
C     TO BACK-SOLVE FOR THE FIRST K X(I) VALUES.
C
      ISUB(K)=KMAX
      COE(K,N+1)=0.
      DO 220 J=KPLUS,N
      JSUB=LOOKUP(KPLUS,J)
      COE(K,JSUB)=-PART(JSUB)/PART(KMAX)
      COE(K,N+1)=COE(K,N+1)+PART(JSUB)*X(JSUB)
  220 CONTINUE
  500 COE(K,N+1)=(COE(K,N+1)-F)/PART(KMAX)+X(KMAX)
C
C     BACK SUBSTITUTE TO OBTAIN NEXT APPROXIMATION TO X.
C
      X(KMAX)=COE(N,N+1)
      IF(N - 1)316,610,316
```

```
  316 CONTINUE
      CALL BACK(N-1,N,X,ISUB,COF,LOOKUP)
  610 IF(M-1)650,650,625
C
C     TEST FOR CONVERGENCE.
C
  625 DO 630 I=1,N
      IF(ABS((TEMP(I)-X(I))/X(I))-RELCON) 630,630,649
  630 CONTINUE
      JTEST=JTEST+1
      IF(JTEST-3)650,725,725
  649 JTEST=1
  650 DO 660 I=1,N
  660 TEMP(I)=X(I)
  700 CONTINUE
  725 IF(IPRINT-1)800,317,800
  317 CONTINUE
      DO 750 K=1,N
      CALL AUXFCN(X,PART(K),K)
  750 CONTINUE
      WRITE(KOUT,751) (PART(K),K=1,N)
  751 FORMAT(//'  FUNCTION VALUES EVALUATED AT FINAL APPROXIMATION FOLLO
     1W   '//(6E20.8))
      GO TO 800
  775 WRITE(KOUT,776)
  776 FORMAT (/20X,  71HMODIFIED JACOBIAN IS SINGULAR. TRY A DIFFERENT I
     1NITIAL APPROXIMATION.    )
  800 RETURN
      END
```

```
      SUBROUTINE BACK (KMIN,N,X,ISUB,COE,LOOKUP)
C
C     THIS SUBROUTINE BACK-SOLVES THE FIRST KMIN ROWS OF A TRIANGULARIZE
C     D LINEAR SYSTEM FOR IMPROVED  X VALUES IN TERMS PREVIOUS ONES.
C
      DIMENSION X(30),ISUB(30),COE(30,31),LOOKUP(30,30)
      DO 200 KK=1,KMIN
      KM=KMIN-KK+2
      KMAX=ISUB(KM-1)
      X(KMAX)=0.0
      DO 100 J=KM,N
      JSUB=LOOKUP(KM,J)
      X(KMAX)=X(KMAX)+COE(KM-1,JSUB)*X(JSUB)
  100 CONTINUE
      X(KMAX)=X(KMAX)+COE(KM-1,N+1)
  200 CONTINUE
      RETURN
      END
/*
/&
```

03/11/69 AUTOFLOW CHART SET - EC311 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES

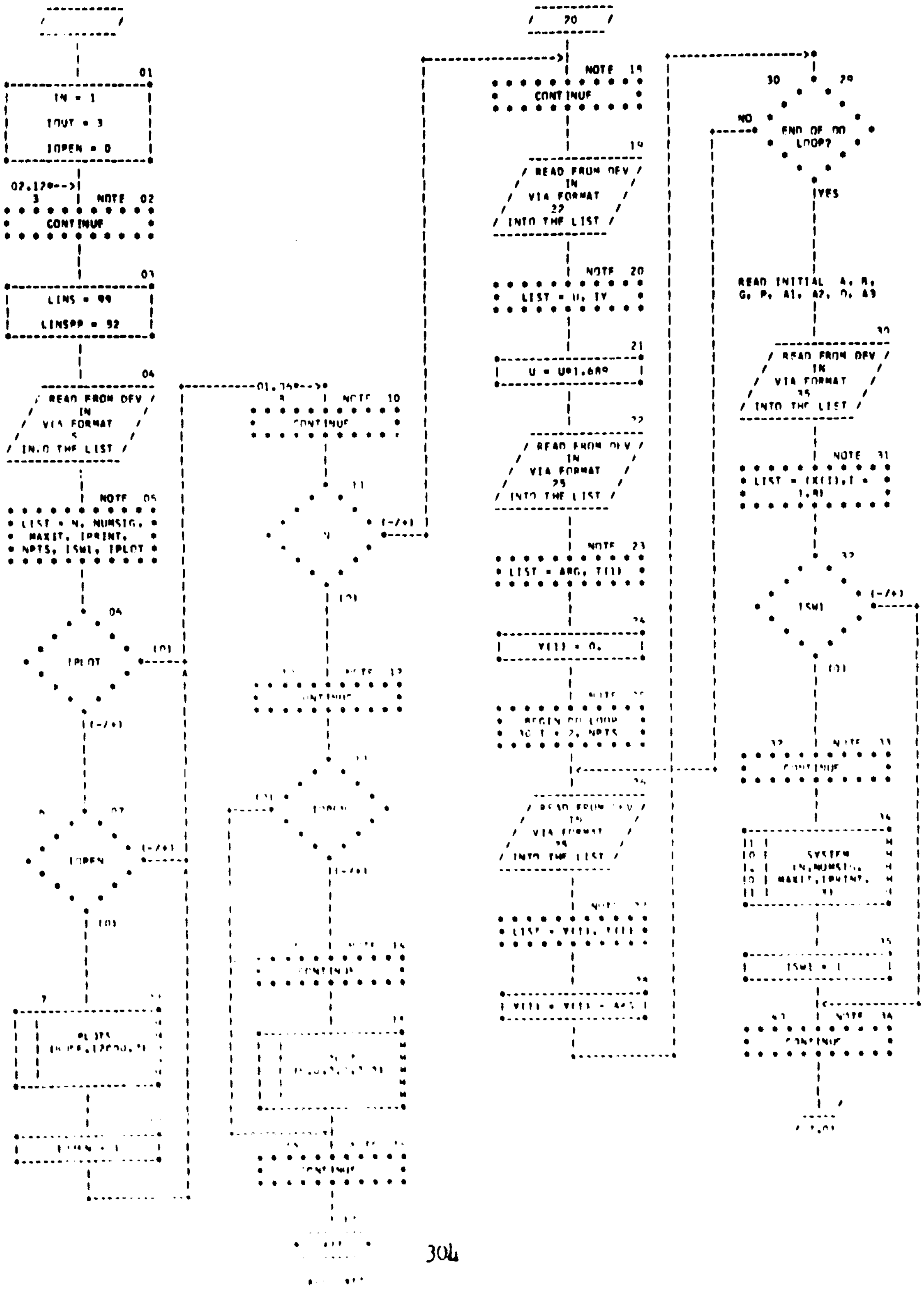


CHART TITLE - PROCEDURES

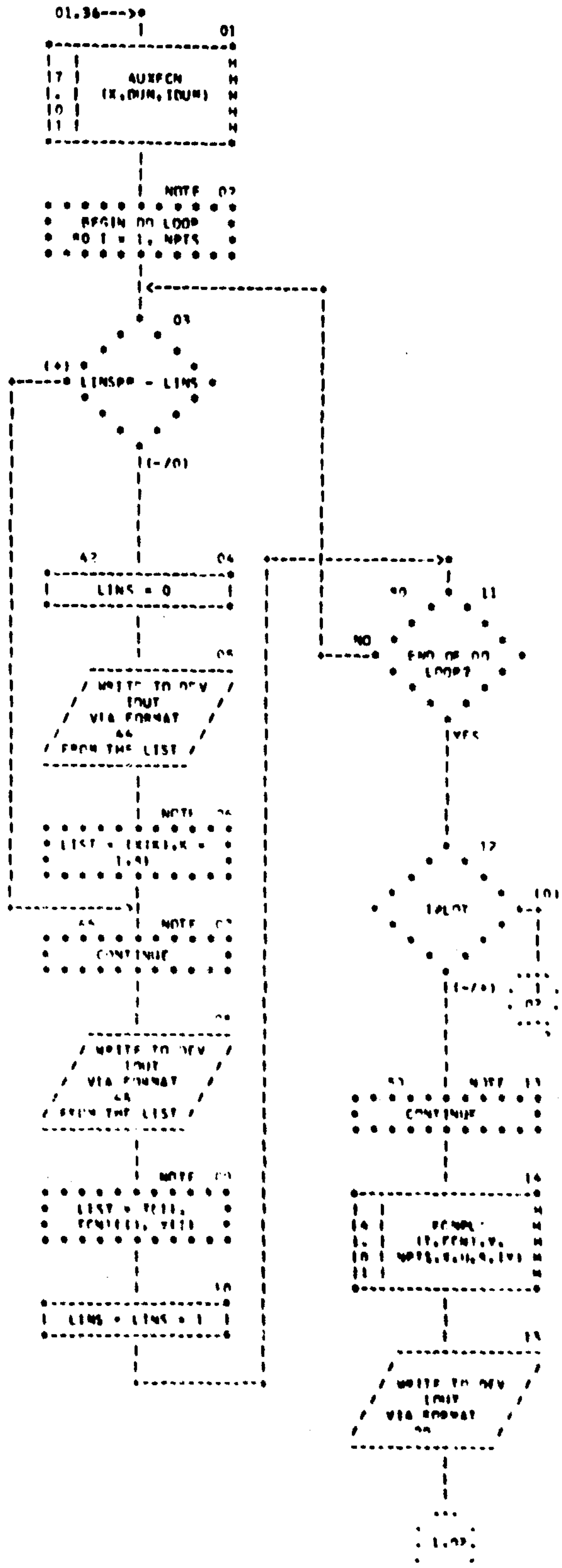
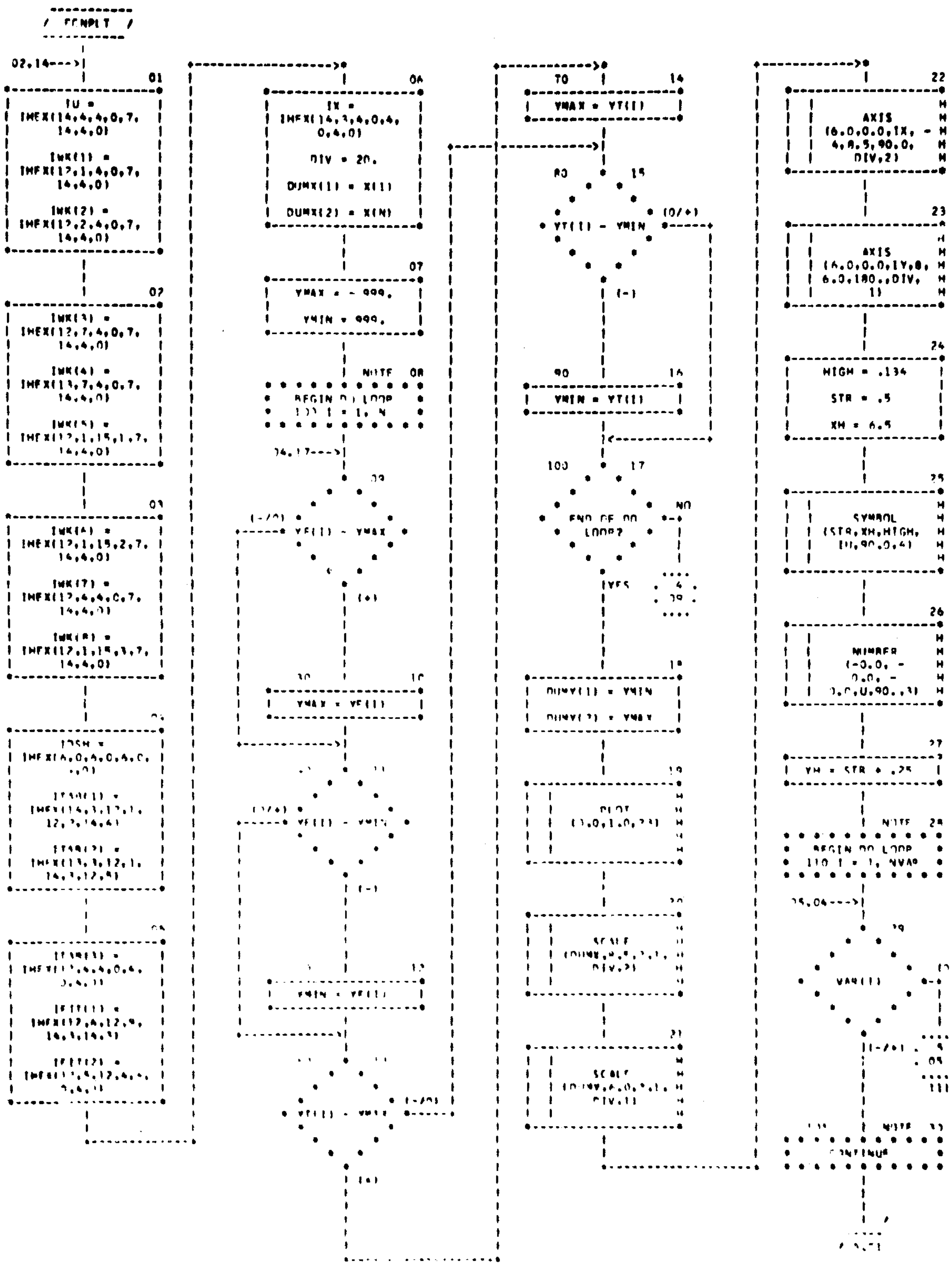


CHART TITLE - SUBROUTINE FCNPLT(X,YF,YT,N,VAR,U,NVAR,IY)



03/11/69 AUTOFLOW CHART SET - EC310 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE FCNPLT(X,YF,YT,N,VAR,U,NVAR,IY)

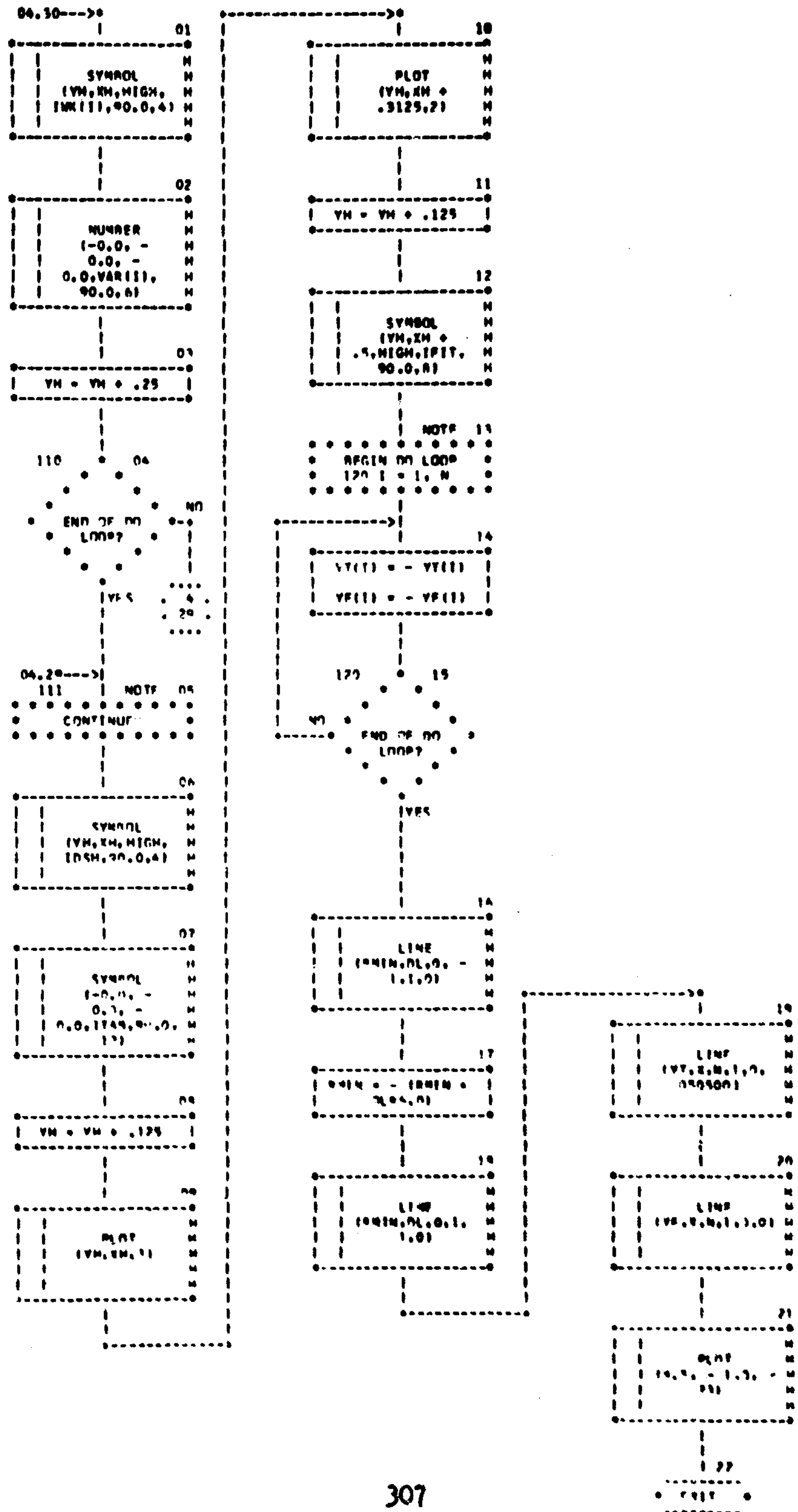
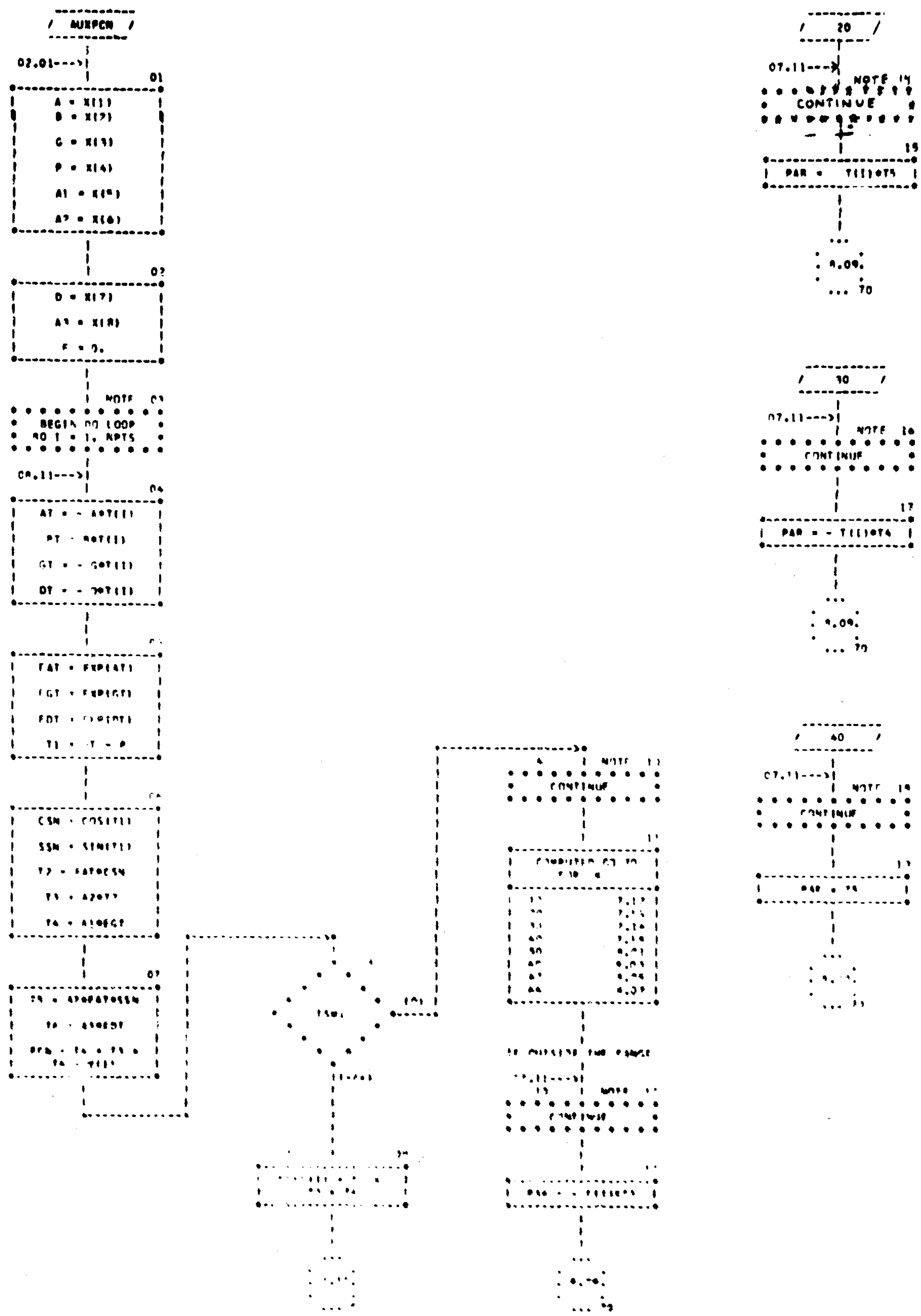
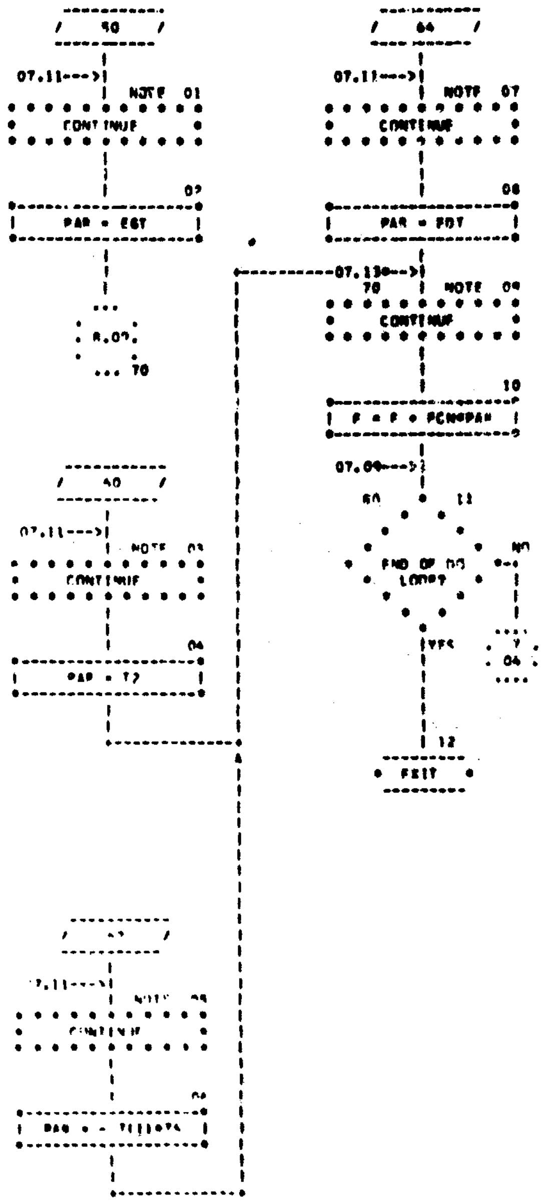


CHART TITLE - SUBROUTINE AUXFCN(X,P,K)



01/11/69 AUTOFLOW CHART SET - FC310 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE AUXFCN(X,F,K)



03/11/69 AUTOFLOW CHART SET - EC310 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE SYSTEM(N,NUMSIG,MAXIT,IPRINT,X)

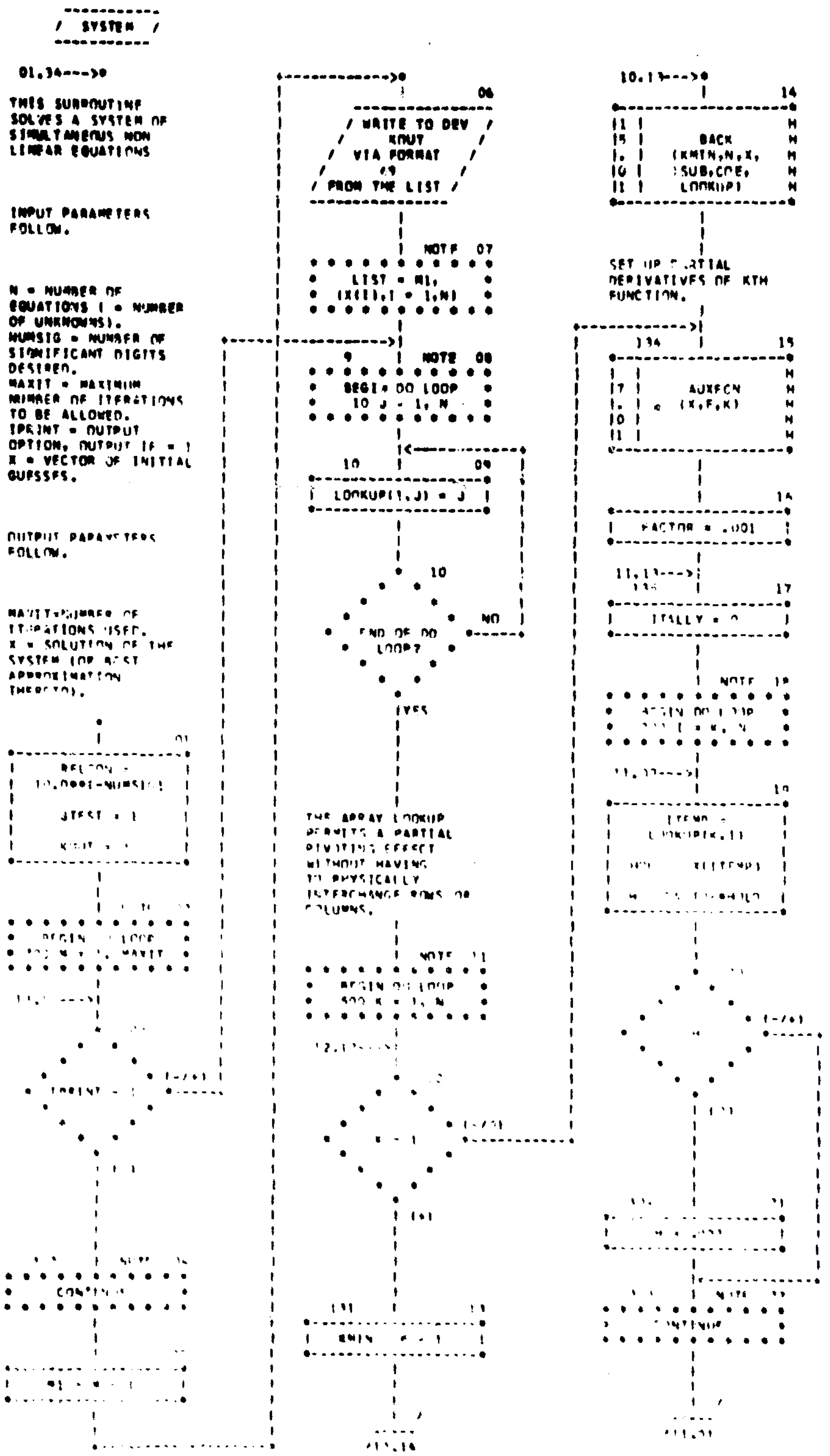
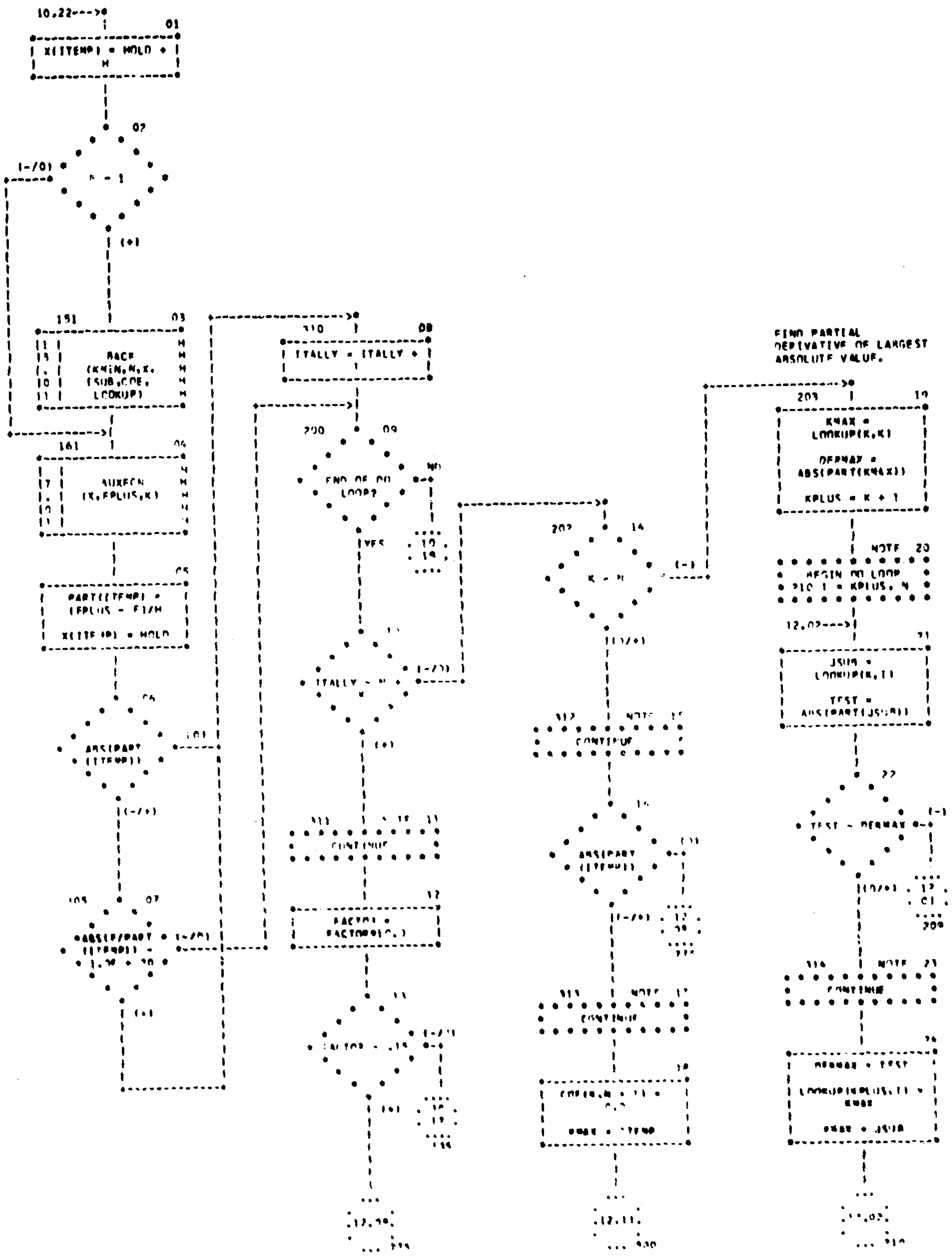
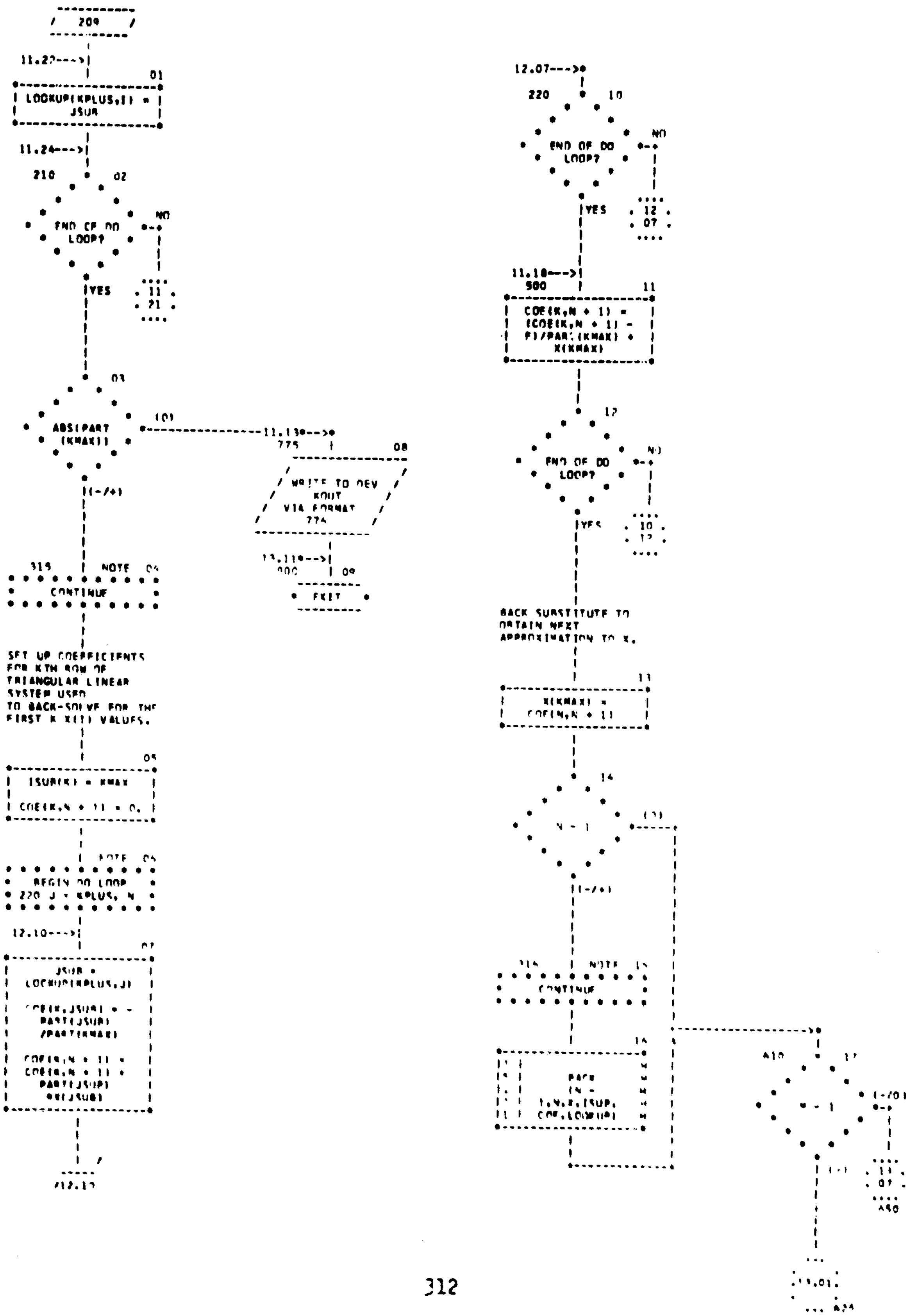


CHART TITLE - SUBROUTINE SYSTEM(N,NUMSIG,MAXIT,IPRINT,K)



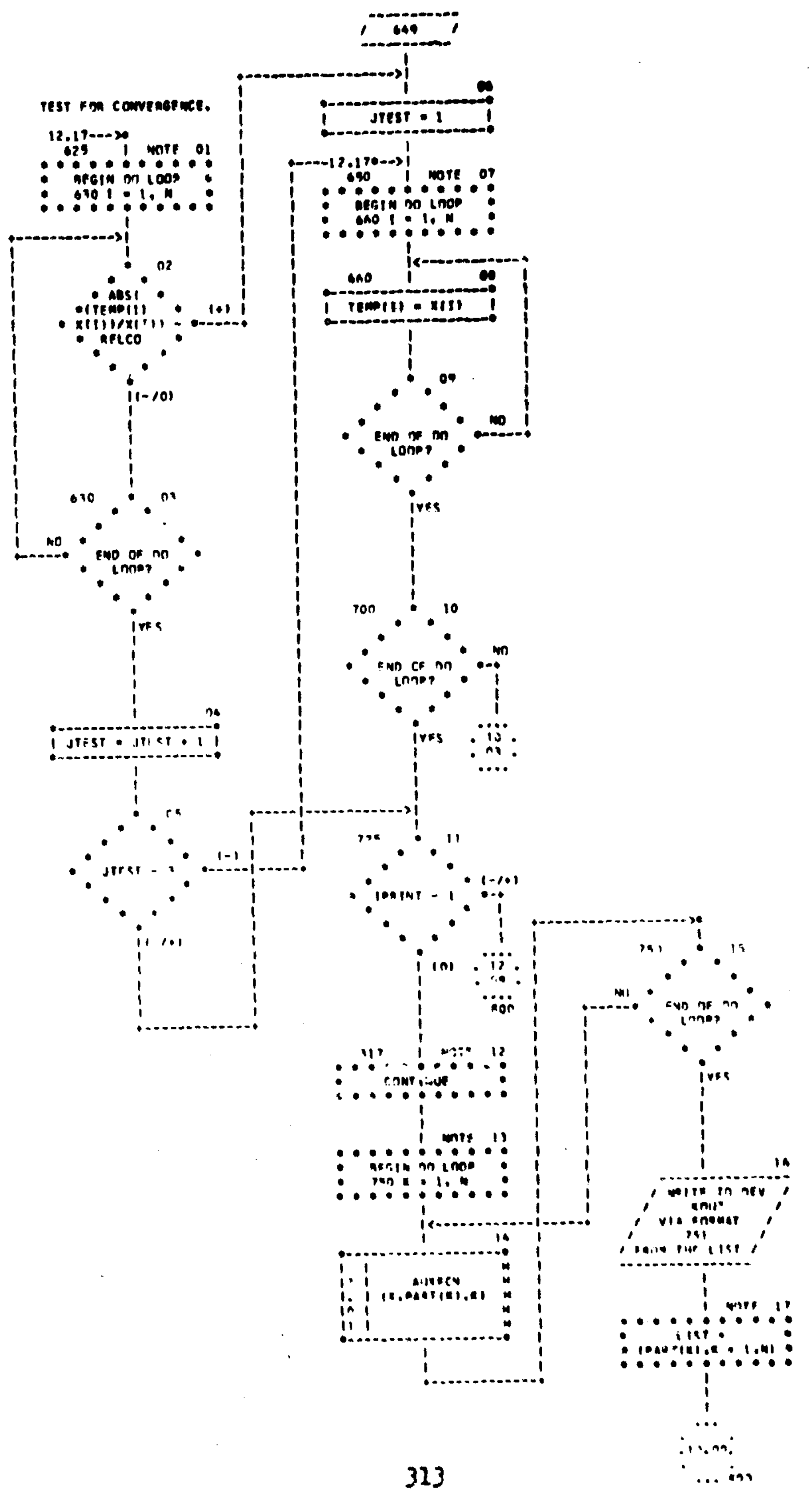
03/11/69 AUTOFLOW CHART SET - EC310 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE SYSTEM(N,NUMSIG,MAXIT,IPRINT,X)



03/11/69 AUTOFLOW CHART SET - EC310 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE SYSTEM(N,NUMSIG,MAXIT,IPRINT,X)



01/11/69 AUTOFLOW CHART SET - EC310 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE BACK(KMIN,N,X,ISUB,COF,LOOKUP)

```
                 ------------
                /   BACK    /
               ------------

          10,140-->*

         THIS SUBROUTINE
         BACK-SOLVES THE FIRST
         KMIN ROWS OF A
         TRIANGULARIZE
         D LINEAR SYSTEM FOR
         IMPROVED  X VALUES IN
         TERMS PREVIOUS ONES.

                    *
                    |  NOTE  01
         * * * * * * * * * * * *
         *    BEGIN DO LOOP    *
         * 200 KK = 1, KMIN    *
         * * * * * * * * * * * *
                    |
                    |<------------------*
                    |            02     |
         *-----------------------*      |
         | KM = KMIN - KK +      |      |
         |          2            |      |
         |                       |      |
         | KMAX = ISUB(KM -      |      |
         |        1)             |      |
         |                       |      |
         |   X(KMAX) = 0.0       |      |
         *-----------------------*      |
                    |                   |
                    |  NOTE  03         |
         * * * * * * * * * * * *        |
         *    BEGIN DO LOOP    *        |
         *    100 J = KM, N    *        |
         * * * * * * * * * * * *        |
                    |                   |
    *------------->|                   |
    |               |            04     |
    |    *-----------------------*      |
    |    |      JSUB =           |      |
    |    |   LOOKUP(KM,J)        |      |
    |    |                       |      |
    |    |     X(KMAX) =         |      |
    |    |     X(KMAX) +         |      |
    |    |     COF(KM -          |      |
    |    | 1,JSUB)*X(JSUB)       |      |
    |    *-----------------------*      |
    |               |                   |
    |               |                   |
    |       100     *      05           |
    |            *     *                |
    |  NO     *           *             |
    *------*   END OF DO     *          |
             *   LOOP?    *             |
                *      *                |
                    *                   |
                    |YES                |
                    |                   |
                    |            06     |
         *-----------------------*      |
         |     X(KMAX) =         |      |
         |     X(KMAX) +         |      |
         | COF(KM - 1,N + 1)     |      |
         *-----------------------*      |
                    |                   |
                    |                   |
       200          *      07           |
                 *     *                |
              *           *      NO     |
           *   END OF DO     *------*
             *   LOOP?    *
                *      *
                    *
                    |YES
                    |
                    |
                    | 08
             ----------
             *  EXIT  *
             ----------
```

```
//        JOB   EC790
//        EXEC FFORTRAN
      DIMENSION Y(13), TL(12)
      DIMENSION SAVE(300,16), ILOC(16), BUFF(3000)
      REAL IX,IY,IZ,IXY,IXZ,IYZ
C
      COMMON H, HMAX, HMIN, DH, FCT, TL, NGS, N, IS1, NPNT
C
      COMMON TLIM, RHOL2, RHOL3, RHOL4, RHOL5, WMB, ETA, ETAM1, ISW2
C
      COMMON          XQQ, XRR, XRP, XUD, XVR ,XWQ, XUU, XVV,
     1                XWW, XDRDR, XDSDS, XDBDB, XVVE, XWWE, XDRDRE, XDSDSE
C
      COMMON          YRD, YPD, YPAP, YPQ, YQR, YVD, YVQ, YWP,
     1                YWR, YR, YP, YARDR, YVAR, YSTR, YV, YVAV,
     2                YVW, YDR, YRE, YVE, YVAVE, YDRE
C
      COMMON          ZQD, ZPP, ZRR, ZRP, ZWD, ZVR, ZVP, ZQ,
     1                ZAQDS, ZWAQ, ZSTR, ZW, ZWAW, ZAW, ZWW, ZVV,
     2                ZDS, ZDB, ZQE, ZWE, ZWAWE, ZDSE
C
      COMMON          AKPD, AKRD, AKQR, AKPQ, AKPAP, AKP, AKR, AKVD,
     1                AKVQ, AKWP, AKWR, AKSTR, AKV, AKVAV, AKVW, AKDR,
     2                AKSTRE
C
      COMMON          AMQD, AMPP, AMRR, AMRP, AMQAQ, AMWD, AMVR, AMVP,
     1                AMQ, AMAQDS, AMAWQ, AMSTR, AMW, AMWAW, AMAW, AMWW,
     2                AMVV, AMDS, AMDB, AMQE, AMWE, AMWAWE, AMDSE
C
      COMMON          ANRD, ANPD, ANPQ, ANQR, ANRAR, ANVD, ANWR, ANWP,
     1                ANVQ, ANP, ANR, ANARDR, ANAVR, ANSTR, ANV, ANVAV,
     2                ANVW, ANDR, ANRE, ANVE, ANVAVE, ANDRE
C
      COMMON           IX, IY, IZ, IXY, IXZ, IYZ
C
      COMMON           CW, CB, UC, XB, YB, ZB
C
      COMMON           DR, DS, DB, RHO, AL, AM
C
      COMMON          DRMAX, ETAHI, ETALO, A11, A12, A13
C
      COMMON          A21, A22, A23, A31, A32, A33
C
      COMMON          XG, YG, ZG
C
      COMMON          ILOC, IPLOT, IRUN, IOPEN, NPLT,IOPT
C
      COMMON          Y
C
      COMMON TIME, R1, DELTMA, SWMAX, R2, DELTMI, DSE, DRE, ICYC, NS,
     1 INTSW
C
   46 CONTINUE
      CALL INPUT
```

```
C
C   COMPUTE RHO * L CONSTANTS
C
      RHOH = RHO * .5
      RHOL2 = RHOH * AL * AL
      RHOL3 = RHOL2 * AL
      RHOL4 = RHOL3 * AL
      RHOL5 = RHOL4 * AL
C
C   WRITE OUT HYDRODYNAMIC COEFFICIENTS
      CALL WRITE
      T = 1.
      XDRDR= RHOL2*XDRDR*T
      XDSDS= RHOL2*XDSDS*T
      XDBDB = RHOL2*XDBDB* T
      A11 = RHOL2* A11 *T
      A12 = RHOL2* A12 *T
      A13 = RHOL2* A13 *T
      A21 = RHOL2 * A21 * T
      A22 = RHOL2 * A22 * T
      A23 = RHOL2 * A23 * T
      A31 = RHOL2 * A31 * T
      A32 = RHOL2 * A32 * T
      A33 = RHOL2 * A33 * T
      XUD = RHOL3 * XUD
      YR =  RHOL3 * YR
      YRD = RHOL4 * YRD * T
      YPD = RHOL4 * YPD * T
      YP = RHOL3 * YP * T
      YV = RHOL2 * YV * T
      YVAV = RHOL2 * YVAV * T
      YDR = RHOL2 * YDR * T
      YVD = RHOL3 * YVD
      ZQ =  RHOL3 * ZQ
      ZQD = RHOL4 * ZQD * T
      ZRR = RHOL4 * ZRR * T
      ZVR = RHOL3 * ZVR * T
      ZSTR = RHOL2 * ZSTR * T
      ZW = RHOL2 * ZW * T
      ZWAW = RHOL2 * ZWAW * T
      ZVV = RHOL2 * ZVV * T
      ZDS  = RHOL2 * ZDS * T
      ZDB = RHOL2 * ZDB * T
      ZWD = RHOL3 * ZWD
      T = IX - RHOL5* AKPD
      T = 1./T
      AKRD = RHOL5 * AKRD * T
      AKP = RHOL4 * AKP * T
      AKVD = RHOL4 * AKVD * T
      AKV = RHOL3 * AKV * T
      AKVAV = RHOL3 * AKVAV * T
      AKPD = T
      T = IY - RHOL5 * AMQD
      T = 1./T
```

```
      AMRP = (IZ - IX + RHOL5 * AMRP) * T
      AMRR = RHOL5 * AMRR * T
      AMWD = RHOL4 * AMWD * T
      AMVR = RHOL4 * AMVR * T
      AMQ = RHOL4 * AMQ * T
      AMAWQ = RHOL4 * AMAWQ * T
      AMSTR = RHOL3 * AMSTR * T
      AMW = RHOL3 * AMW * T
      AMWAW = RHOL3 * AMWAW * T
      AMVV = RHOL3 * AMVV * T
      AMDS = RHOL3 * AMDS * T
      AMDB = RHOL3 * AMDB * T
      AMQD = T
      T = IZ-RHOL5*ANRD
      T =  1./T
      ANPQ =(IX-IY+RHOL5*ANPQ)*T
      ANPD = RHOL5 * ANPD * T
      ANVD = RHOL4 * ANVD * T
      ANP = RHOL4 * ANP * T
      ANR  = RHOL4 * ANR * T
      ANV  = RHOL3 * ANV * T
      ANVAV = RHOL3 * ANVAV * T
      ANDR  = RHOL3 * ANDR * T
      ANRD = T
      ZB = CB * ZP
      CALL WRITE
      IPCH = 2
      WRITE(IPCH,80) XDRDR, XDSDS, XDBDB, A11, A12, A13
      WRITE(IPCH,80) A21, A22, A23, A31, A32, A33
      WRITE(IPCH,80) XUD, YR, YRD, YPD, YP, YV
      WRITE(IPCH,80) YVAV, YDR, YVD, ZQ, ZQD, ZRR
      WRITE(IPCH,80) ZVR, ZSTR, ZW, ZWAW, ZVV, ZDS
      WRITE(IPCH,80) ZDB, ZWD, AKPD, AKP, AKVD, AKV
      WRITE(IPCH,80) AKVAV, AKPD, AMRP, AMRR, AMWD, AMVR
      WRITE(IPCH,80) AMQ, AMAWQ, AMSTR, AMW, AMWAW, AMVV
      WRITE(IPCH,80) AMDS, AMDB, AMQD, ANPQ, ANPD, ANVD
      WRITE(IPCH,80) ANP, ANR, ANV, ANVAV, ANDR, ANRD
      WRITE(IPCH,80) DRMAX, ETAHI, ETALO, CW, CB, XG
      WRITE(IPCH,80) ZG, AL, AM, DR, DS, DB
      WRITE(IPCH,80) ZB, UC, TIME, R1, DELTMA, SWMAX
      WRITE(IPCH,80) R2, DELTMI, DSF, DRF,(Y(I),I=1,12)
   80 FORMAT(6F13.6)
      GO TO 45
      END
```

```
      SUBROUTINE WRITE
      DIMENSION Y(13), TL(12)
      DIMENSION SAVE(300,16), ILOC(16), BUFF(3000)
      REAL IX,IY,IZ,IXY,IXZ,IYZ
C
      COMMON H, HMAX, HMIN, DH, FCT, TL, NGS, N, IS1, NPNT
C
      COMMON TLIM, RHOL2, RHOL3, RHOL4, RHOL5, WMB, ETA, ETAM1, ISW2
C
      COMMON        XQQ, XPR, XRP, XUD, XVR ,XWQ, XUU, XVV,
     1              XWW, XDRDR, XDSDS, XDBDB, XVVE, XWWE, XDRDRE, XDSDSE
C
      COMMON        YRD, YPD, YPAP, YPQ, YQR, YVD, YVQ, YWP,
     1              YWP, YP, YP, YARDR, YVAR, YSTR, YV, YVAV,
     2              YVW, YDR, YRE, YVE, YVAVE, YDRE
C
      COMMON        ZQD, ZPP, ZRR, ZRP, ZWD, ZVR, ZVP, ZQ,
     1              ZAQDS, ZWAQ, ZSTR, ZW, ZWAW, ZAW, ZWW, ZVV,
     2              ZDS, ZDB, ZQE, ZWE, ZWAWE, ZDSE
C
      COMMON        AKPD, AKRD, AKQR, AKPQ, AKPAP, AKP, AKR, AKVD,
     1              AKVQ, AKWP, AKWR, AKSTR, AKV, AKVAV, AKVW, AKDR,
     2              AKSTRE
C
      COMMON        AMQD, AMPP, AMRR, AMRP, AMQAQ, AMWD, AMVR, AMVP,
     1              AMQ, AMAQDS, AMAWQ, AMSTR, AMW, AMWAW, AMAW, AMWW,
     2              AMVV, AMDS, AMDB, AMQE, AMWE, AMWAWE, AMDSE
C
      COMMON        ANPD, ANPD, ANPQ, ANQR, ANRAR, ANVD, ANWR, ANWP,
     1              ANVQ, ANP, ANR, ANARDR, ANAVR, ANSTR, ANV, ANVAV,
     2              ANVW, ANDR, ANRE, ANVE, ANVAVE, ANDRE
C
      COMMON         IX, IY, IZ, IXY, IXZ, IYZ
C
      COMMON         CW, CB, UC, XB, YB, ZB
C
      COMMON         DP, DS, DR, RHO, AL, AM
C
      COMMON        DRMAX, ETAHI, ETALO, A11, A12, A13
C
      COMMON        A21, A22, A23, A31, A32, A33
C
      COMMON        XG, YG, ZG
C
      COMMON        ILOC, IPLOT, IRUN, IOPEN, NPLT,IOPT
C
      COMMON        Y
C
      COMMON TIME, R1, DELTMA, SWMAX, R2, DELTMI, DSE, DRE, ICYC, NS,
     1 INTSW
C
C
      IOUT = 3
   25 FORMAT(1H1,50X,'DATA PREPARATION'//)
```

```
      WRITE(IOUT,24)
      WRITE(IOUT,1) XQQ, YRD, ZQD, AKPD, AMQD, ANRD, XPR, YPD, ZPP,
     1 AKRD, AMPP, ANPD, XRP, YPAP, ZRR, AKQR, AMRR, ANPQ, XUD,
     2 YPQ, ZRP, AKPQ, AMPP, ANQR,  XVR, YQR, ZWD, AKPAP, AMQAQ,
     3 ANRAR, XWQ, YVD, ZVR, AKP, AMWD, ANVD
    1 FORMAT(1H ,'XQQ',4X,F12.5,'  YRD',4X,F12.5,'  ZQD',4X,F12.5,
     1'  KPD',4X,F12.5,'  MQD',4X,E12.5,'  NRD',4X,F12.5/1H ,
     2'XPR',4X,F12.5,'  YPD',4X,F12.5,'  ZPP',4X,F12.5,'  KPD',4X,
     3F12.5,'  MPP',4X,F12.5,'  NPD',4X,F12.5/1H ,
     4'XRP',4X,E12.5,'  YPAP',3X,F12.5,'  ZRR',4X,F12.5,'  KQR',4X,
     5F12.5,'  MRR',4X,F12.5,'  NPQ',4X,E12.5/1H ,
     6'XUD',4X,F12.5,'  YPQ', 4X,F12.5,'  ZRP',4X,F12.5,'  KPQ',4X,
     7E12.5,'  MRP',4X,F12.5,'  NQR',4X,F12.5/1H ,
     8'XVR',4X,E12.5,'  YQR',4X,F12.5,'  ZWD',4X,F12.5,'  KPAP',3X,
     9F12.5,'  MQAQ',3X,F12.5,'  NRAR',3X,F12.5/1H ,
     A'XWQ',4X,E12.5,'  YVD',4X,F12.5,'  ZVR',4X,E12.5,'  KP',5X,
     BE12.5,'  MWD',4X,F12.5,'  NVD',4X,F12.5)
      WRITE(IOUT,11)  XUU, YVQ, ZVP, AKR,
     1 AMVR, ANWR, XVV, YWP, ZQ, AKVD, AMVP, ANWP
   11 FORMAT(1H ,'XUU',4X,F12.5,'  YVQ',4X,E12.5,'  ZVP',4X,F12.5,
     1'  KR',5X,F12.5,'  MVR',4X,F12.5,'  NWR',4X,F12.5/1H ,
     2'XVV',4X,E12.5,'  YWP',4X,F12.5,'  ZQ',5X,F12.5,'  KVD',4X,
     3F12.5,'  MVP',4X,F12.5,'  NWP',4X,E12.5)
      WRITE(IOUT,2) XWW, YWR, ZAQDS, AKVQ, AMQ, ANVQ, XDRDR, YR,
     1ZWAQ, AKWP, AMAQDS, ANP, XDSDS, YP, ZSTR, AKWR, AMAWQ, ANR,
     2XDBDB, YARDR, ZW, AKSTR, AMSTR, ANARDR,XVVF,YVAR, ZWAW, AKV,
     3AMW, ANAVR,XWWF,YSTR,ZAW,AKVAV,AMWAW,ANSTR
    2 FORMAT(1H ,'XWW',4X,F12.5,'  YWR',4X,F12.5,'  ZAQDS',2X,F12.5,
     1'  KVQ',4X,F12.5,'  MQ',5X,E12.5,'  NVQ',4X,E12.5/1H ,
     2'XDRDR',2X,F12.5,'  YR',5X,F12.5,'  ZWAQ',3X,F12.5,'  KWP',4X,
     3F12.5,'  MAQDS',2X,F12.5,'  NP',5X,E12.5/1H ,
     4'XDSDS',2X,F12.5,'  YP',5X,F12.5,'  ZSTR',3X,F12.5,'  KWR',4X,
     5F12.5,'  MAWQ',3X,F12.5,'  NR',5X,F12.5/1H ,
     6'XDBDB',2X,F12.5,'  YARDR',2X,F12.5,'  ZW',5X,F12.5,'  KSTR',3X,
     7E12.5,'  MSTR',3X,F12.5,'  NARDR',2X,E12.5/1H ,
     8'XVVF',3X,F12.5,'  YVAR',3X,F12.5,'  ZWAW',3X,F12.5,'  KV',5X,
     9F12.5,'  MW',5X,F12.5,'  NAVR',3X,E12.5/1H ,
     A'XWWF',3X,F12.5,'  YSTR',3X,F12.5,'  ZAW',4X,F12.5,'  KVAV',3X,
     BE12.5,'  MWAW',3X,F12.5,'  NSTR',3X,F12.5)
      WRITE(IOUT,22)                                 XDRDRF, YV, ZWW,
     1AKVW, AMAW, ANV, XDSDSF, YVAV, ZVV, AKQR, AMWW, ANVAV
   22 FORMAT(1H ,'XDRDRF',1X,F12.5,'  YV',5X,F12.5,'  ZWW',4X,F12.5,
     1'  KVW',4X,F12.5,'  MAW',4X,F12.5,'  NV',5X,F12.5/1H ,
     2'XDSDSF',1X,F12.5,'  YVAV',3X,F12.5,'  ZVV',4X,F12.5,'  KQR',4X,
     3F12.5,'  MWW',4X,F12.5,'  NVAV',3X,F12.5)
      WRITE(IOUT,3)         YVW, ZDS, AKSTRF, AMVV, ANVW,        YDR,
     1ZDB, AMDS, ANDR,              YRF, ZQF, AMDR, ANRF, YVF, ZWF, AMQF,
     2ANVF, YVAVF, ZWAWF, AMWF, ANVAVF, YDRF, ZDSF, AMWAWF, ANDRF, AMDSF
    3 FORMAT(1H ,   19X,            '  YVW',4X,F12.5,'  ZDS',4X,F12.5,
     1'  KSTRF',2X,F12.5,'  MVV',4X,F12.5,'  NVW',4X,F12.5/1H ,
     219X,                '  YDR',4X,F12.5,'  ZDB',4X,F12.5,21X,
     3'  MDS',4X,F12.5,'  NDR',4X,F12.5/1H ,
     419X,                '  YRF'           ,4X,F12.5,'  ZQF',4X,F12.5,21X,
     5'  MDR',4X,F12.5,'  NRF',4X,F12.5/1H ,
```

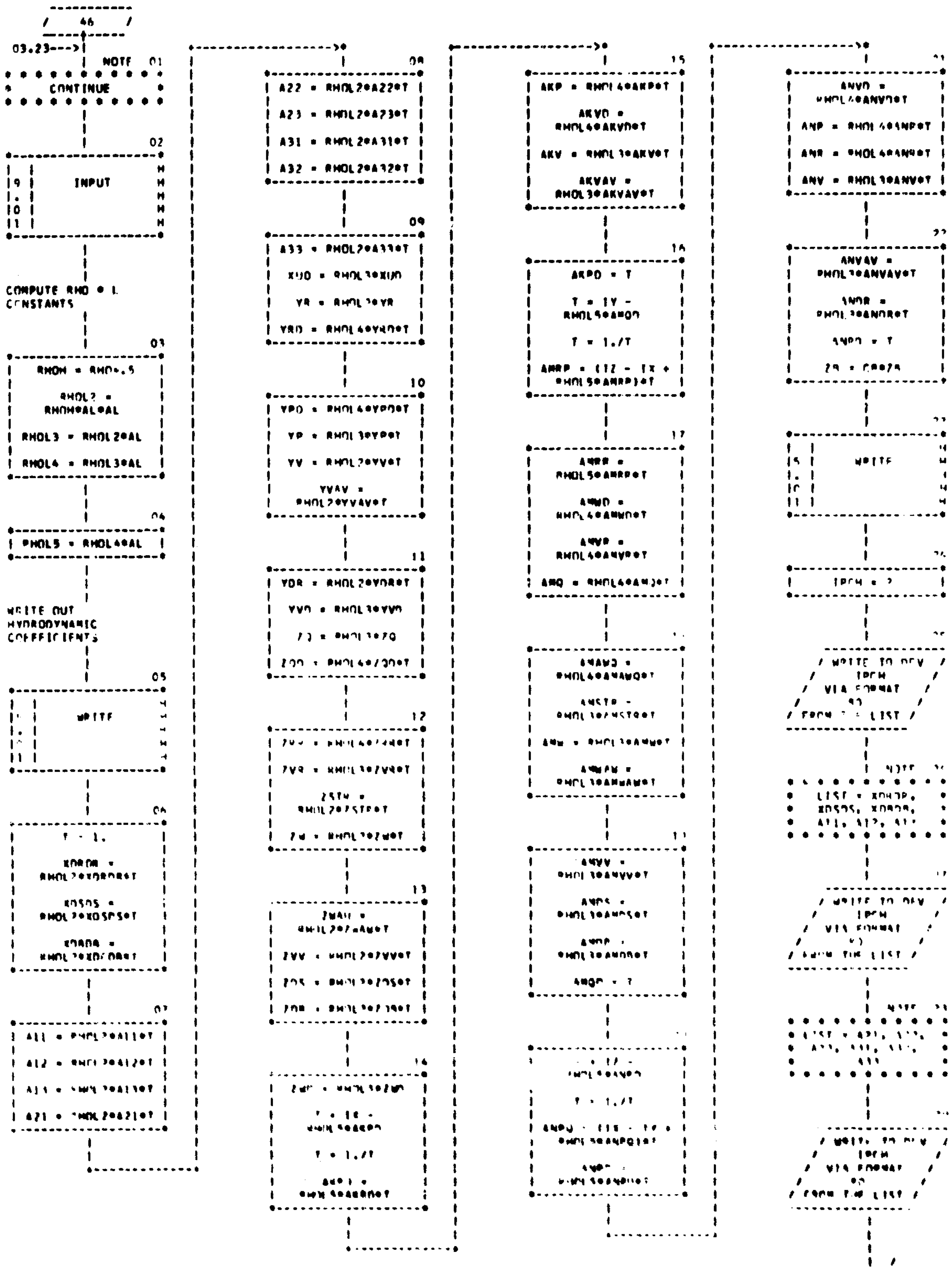
```
6 19X,'  YVF',4X,F12.5,'  ZWF',4X,E12.5,21X,'  MQF',4X,F12.5,
7'  NVF',4X,E12.5/1H , 19X,'  YVAVF',2X,F12.5,'  ZWAWF',2X,F12.5,
8 21X,'  MWF',4X,F12.5,'  NVAVF',2X,F12.5/1H ,19X,'  YDRF',3X,
9 F12.5,'  ZDSF',3X,F12.5,21X,'  MWAWF',2X,F12.5,'  NDRF',3X,
A E12.5,/1H ,84X,'MDSF',3X,F12.5//)
 WRITE(IOUT,4) IX, IY, IZ, IXY, IXZ, IYZ, CW, CB, UC, XB, YB,
1 ZB, DR, DS, DB, RHO, AL, AM
4 FORMAT(1H ,'IX',5X,F12.5,'  IY',5X,F12.5,'  IZ',5X,F12.5,
1'  IXY',4X,E12.5,'  IXZ',4X,F12.5,'  IYZ',4X,F12.5/1H ,
2'W',6X,E12.5,'  B',6X,F12.5,'  UC',5X,E12.5,'  XB',5X,F12.5,
3'  YB',5X,F12.5,'  ZB',5X,F12.5/1H ,
4'DR',5X,F12.5,'  DS',5X,F12.5,'  DB',5X,F12.5,'  RHO',4X,F12.5,
5'  L',6X,F12.5,'  M'6X,F12.5 )
 WRITE(IOUT,6) A11, A21, A31, DRMAX, ETAHI, ETALO, A12, A22, A32,
1 XG, YG, ZG, A13, A23, A33
2, H,INTSW, TIME, R1, DELTMA, SWMAX, R2, DELTMI, DSF, DRF, ICYC, NS
6 FORMAT(1H ,'A11',4X,F12.5,'  A21',4X,E12.5,'  A31',4X,F12.5,
1'  DRMAX',2X,F12.5,'  ETAHI',2X,F12.5,'  ETALO'.2X,F12.5/
21H ,'A12',4X,F12.5,'  A22',4X,F12.5,'  A32',4X,F12.5,
3'  XG',5X,F12.5,'  YG',5X,F12.5,'  ZG',5X,F12.5/
41H ,'A13',4X,F12.5,'  A23',4X,F12.5,'  A33',4X,F12.5,
5'  H',6X,F12.5,'  INTSW',2X,I2/
61H ,'TIME'3X,F12.5,'  R1',5X,F12.5,'  DELTMA',1X,F12.5,'  SWMAX',
72X,F12.5,'  R2',5X,F12.5,'  DELTMI',1X,F12.5/
81H ,'DSF',4X,F12.5,'  DRF',4X,F12.5,'  ICYC',3X,I2,10X,
9'  NS',5X,I2/)
 WRITE(IOUT,5) IRUN
5 FORMAT(1H ,'RUN NO  ', I5/)
C
  RETURN
  END
```

```
      SUBROUTINE INPUT
      DIMENSION Y(13), TL(12), ILOC(16), YHOLD(13), COM(219)
      EQUIVALENCE (COM(1),H)
      REAL IX,IY,IZ,IXY,IXZ,IYZ
C
      COMMON H, HMAX, HMIN, DH, FCT, TL, NGS, N, IS1, NPNT
C
      COMMON TLIM, RHOL2, RHOL3, RHOL4, RHOL5, WMB, ETA, ETAM1, ISW2
C
      COMMON       XQQ, XRR, XRP, XUD, XVR ,XWQ, XUU, XVV,
     1             XWW, XDRDR, XDSDS, XDBDB, XVVE, XWWE, XDRDRE, XDSDSE
C
      COMMON       YRD, YPD, YPAP, YPQ, YQR, YVD, YVQ, YWP,
     1             YWR, YR, YP, YARDR, YVAR, YSTR, YV, YVAV,
     2             YVW, YDR, YPE, YVE, YVAVE, YDRE
C
      COMMON       ZQD, ZPP, ZRR, ZRP, ZWD, ZVR, ZVP, ZQ,
     1             ZAQDS, ZWAQ, ZSTP, ZW, ZWAW, ZAW, ZWW, ZVV,
     2             ZDS, ZDB, ZQE, ZWE, ZWAWE, ZDSE
C
      COMMON       AKPD, AKRD, AKQR, AKPQ, AKPAP, AKP, AKR, AKVD,
     1             AKVQ, AKWP, AKWR, AKSTR, AKV, AKVAV, AKVW, AKDR,
     2             AKSTRE
C
      COMMON       AMQD, AMPP, AMRR, AMRP, AMQAQ, AMWD, AMVR, AMVP,
     1             AMQ, AMAQDS, AMAWQ, AMSTR, AMW, AMWAW, AMAW, AMWW,
     2             AMVV, AMDS, AMDB, AMQE, AMWE, AMWAWE, AMDSE
C
      COMMON       ANRD, ANPD, ANPQ, ANQR, ANRAR, ANVD, ANWR, ANWP,
     1             ANVQ, ANP, ANR, ANARDR, ANAVR, ANSTR, ANV, ANVAV,
     2             ANVW, ANDR, ANRE, ANVE, ANVAVE, ANDRE
C
      COMMON        IX, IY, IZ, IXY, IXZ, IYZ
C
      COMMON        CW, CB, UC, XB, YB, ZB
C
      COMMON        DP, DS, DB, RHO, AL, AM
C
      COMMON       DRMAX, ETAHI, ETALO, A11, A12, A13
C
      COMMON       A21, A22, A23, A31, A32, A33
C
      COMMON       XG, YG, ZG
C
      COMMON       ILOC, IPLOT, IRUN, IOPEN, NPLT,IOPT
C
      COMMON       Y
C
      COMMON TIME, R1, DELTMA, SWMAX, R2, DELTMI, DSE, DRE, ICYC, NS,
     1 INTSW
C
      IN  = 1
      READ(IN,50) NGS, NPNT, IPLOT, IRUN, NPLT, IOPT, ICYC, NS,INTSW
      IF(IRUN)70,60,70
```

```
   60 CALL EXIT
   70 CONTINUE
      READ(IN,50) (ILOC(I), I = 1, 16)
   50 FORMAT(16I5)
      READ(IN,100)TO,HO,DH,HMAX,HMIN,FCT,TLIM
      H = HO
  100 FORMAT(8F10.5)
      READ(IN,100)(TL(I),I=1,12)
      READ(IN,100)(Y(I), I=1,12)
      Y(13)= TO
C
      READ(IN,100) XQQ, XRR, XRP, XUD, XVR ,XWQ, XUU, XVV,
     1XWW, XDRDR, XDSDS, XDBDB, XVVE, XWWE, XDRDRE, XDSDSE
C
      READ(IN,100) YRD, YPD, YPAP, YPQ, YQR, YVQ, YVQ, YWP,
     1             YWR, YP, YR, YAPDR, YVAR, YSTR, YV, YVAV,
     2             YVW, YDR, YRE, YVE, YVAVE, YDRE
C
      READ(IN,100) ZQD, ZPP, ZRR, ZRP, ZWD, ZVR, ZVP, ZQ,
     1             ZAQDS, ZWAQ, ZSTR, ZW, ZWAW, ZAW, ZWW, ZVV,
     2             ZDS, ZDB, ZQE, ZWE, ZWAWE, ZDSE
C
      READ(IN,100) AKPD, AKRD, AKQR, AKPQ, AKPAP, AKP, AKR, AKVD,
     1             AKVQ, AKWP, AKWP, AKSTR, AKV, AKVAV, AKVW, AKDP,
     2             AKSTRE
C
      READ(IN,100) AMQD, AMPP, AMRR, AMRP, AMQAQ, AMWD, AMVR, AMVP,
     1             AMQ, AMAQDS, AMAWQ, AMSTR, AMW, AMWAW, AMAW, AMWW,
     2             AMVV, AMDS, AMDB, AMQE, AMWE, AMWAWE, AMDSE
C
      READ(IN,100) ANRD, ANPD, ANPQ, ANQR, ANPAR, ANVD, ANWR, ANWP,
     1             ANVQ, ANP, ANR, ANARDR, ANAVR, ANSTR, ANV, ANVAV,
     2               VW, ANDR, ANRE, ANVE, ANVAVE, ANDRE
C
      READ(IN,100)  IX, IY, IZ, IXY, IXZ, IYZ
C
      READ(IN,100)  CW, CB, HC, XB, YB, ZB
C
      READ(IN,100)  DR, DS, DB, RHO, AL, AM
C
      READ (IN,100) DRMAX, ETAHI, ETALO, A11, A12, A13
C
      READ(IN,100)  A21, A22, A23, A31, A32, A33
C
      READ(IN,100) XG, YG, ZG
C
      READ(IN,100) TIME, R1, DELTMA, SWMAX, R2, DELTMI, DSE, DRE
C
      RETURN
      END
/*
/&
```

03/24/69 AUTOFLOW CHART SET - EC790 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES



03/24/69 AUTOFLOW CHART SET - EC790 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES

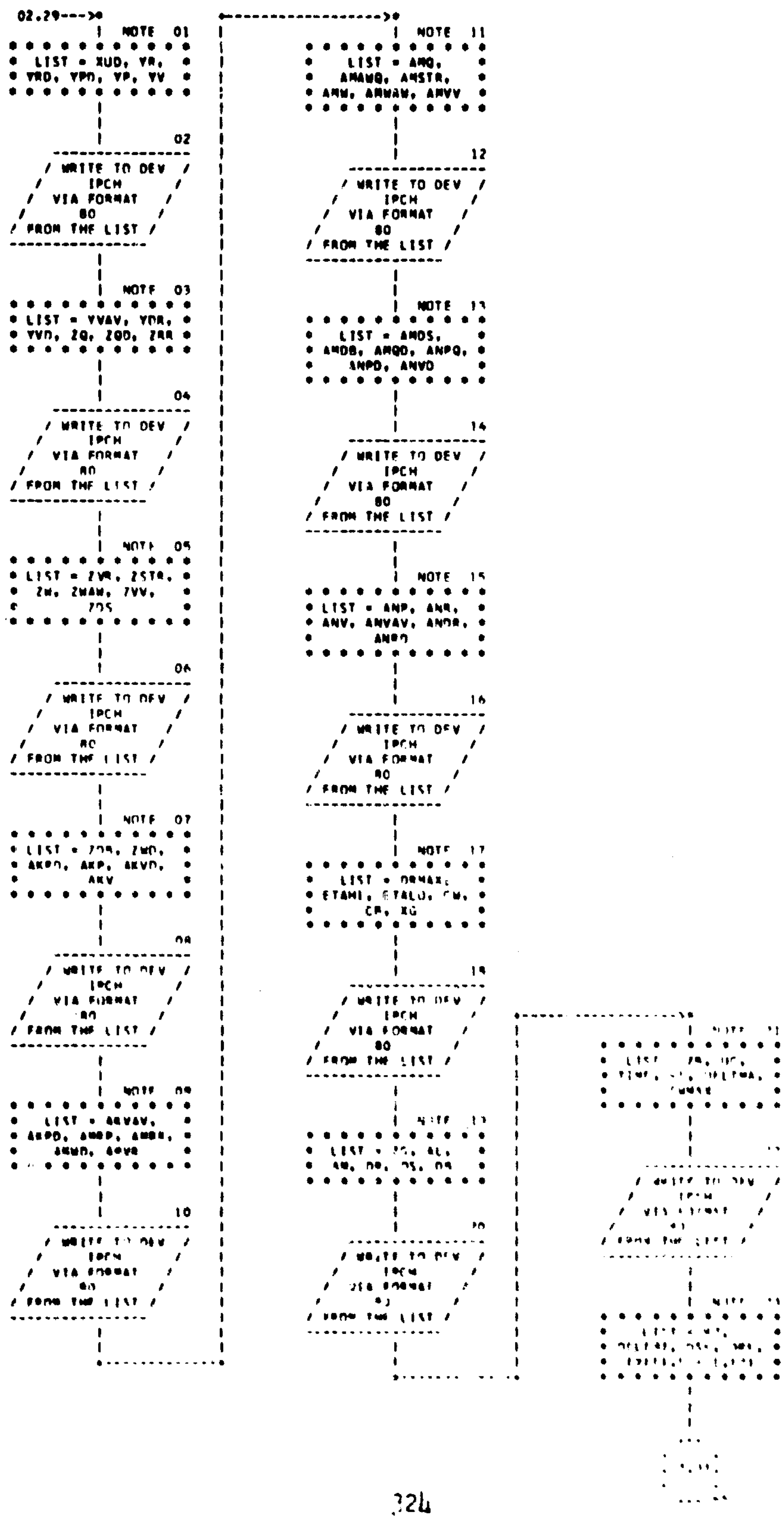
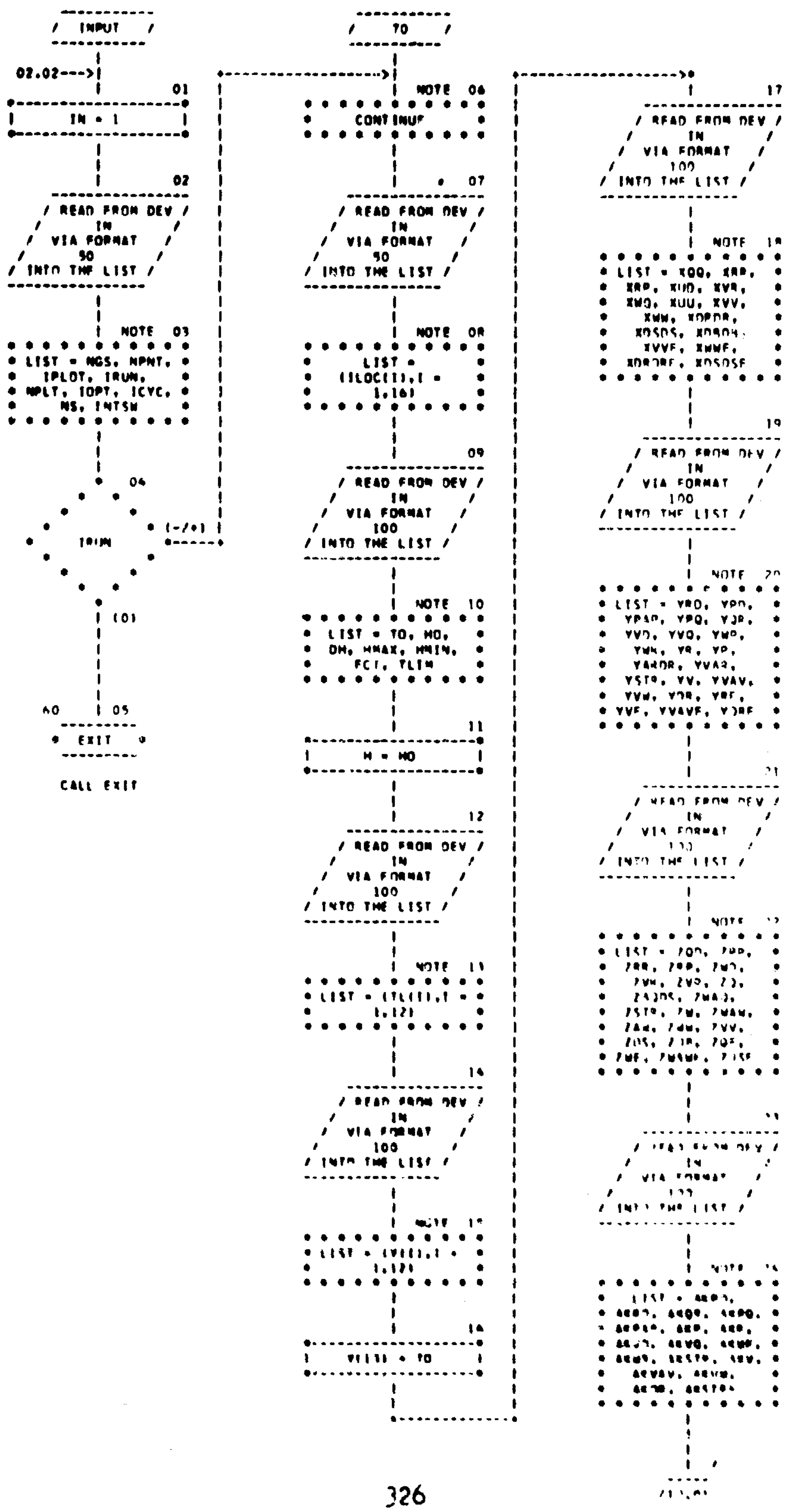


CHART TITLE - SUBROUTINE WRITE

```
/ WRITE /

01.050--> 

01
IOUT = 3

02
/ WRITE TO DEV
IOUT
VIA FORMAT
24 /

03
/ WRITE TO DEV
IOUT
VIA FORMAT
1
FROM THE LIST /

NOTE 04
LIST = XQQ, YRQ,
ZQD, AKPD, ANQD,
ANRD, XRR, YPD,
ZPP, AKRD, ANPP,
ANPD, XRP, YPAP,
ZRR, AKQR, ANRR,
ANPQ, XUD, YPQ,
ZRP, AKPQ, ANRP,
ANQR, XVR, YQR,
ZWD, AKPAP,
ANQAQ, ANRAR,
XWQ, YVD, ZVR,
AKP, ANWD, ANVO

05
/ WRITE TO DEV
IOUT
VIA FORMAT
11
FROM THE LIST /

NOTE 06
LIST = XUU, YVO,
ZVP, AKR, ANVP,
ANWR, XVV, YWP,
ZQ, AKVD, ANVP,
ANWP

07
/ WRITE TO DEV
IOUT
VIA FORMAT
7
FROM THE LIST /

NOTE 08
LIST = XWW, YWR,
ZAQDS, AKVQ, ANQ,
ANVQ, XDRDR, YR,
ZWAQ, AKWP,
ANAQDS, ANP,
XDSDS, YP, ZSTR,
AKWR, ANAWQ, ANR,
XDBDB, YARDR, ZW,
AKSTR, ANSTR,
ANARDR, XVVE,
YVAR, ZWAW, AKV,
ANW, ANAVR, XWWE,
YSTR, ZAW, AKVAV,
ANWAW, ANSTR

09
/ WRITE TO DEV
IOUT
VIA FORMAT
22
FROM THE LIST /

NOTE 10
LIST = XDRDRE,
YV, ZWW, AKVW,
ANAW, ANV,
XDSDSE, YVAV,
ZVV, AKDR, ANWW,
ANVAV

11
/ WRITE TO DEV
IOUT
VIA FORMAT
3
FROM THE LIST /

NOTE 12
LIST = YVW, ZDS,
AKSTRE, ANVV,
ANVW, YDR, ZDR,
ANDS, ANDR, YRE,
ZQE, ANDB, ANRE,
YVE, ZWE, ANQE,
ANVE, YVAVE,
ZWAWE, ANWE,
ANVAVE, YDRE,
ZDSE, ANWAWE,
ANDRE, ANDSE

13
/ WRITE TO DEV
IOUT
VIA FORMAT
6
FROM THE LIST /

NOTE 14
LIST = IX, IY,
IZ, IXY, IXZ,
IYZ, CW, CR, UC,
XB, YB, ZB, DR,
DS, TA, OMD, AL,
AM

15
/ WRITE TO DEV
IOUT
VIA FORMAT
4
FROM THE LIST /

NOTE 16
LIST = A11, A21,
A31, ODMAX,
ETAMI, ETALO,
A12, A22, A32,
XG, YG, ZG, A13,
A23, A33, W,
[illegible], TIME, R1,
DELTMA, SUMAX,
R2, DELTMI, [illegible],
[illegible], [illegible], [illegible]

17
/ WRITE TO DEV
IOUT
VIA FORMAT
5
FROM THE LIST /

NOTE 18
LIST = [illegible]

19
EXIT
```

03/24/69 AUTOFLOW CHART SET - EC790 NAVTRADEVCEN 68-C-0050-2

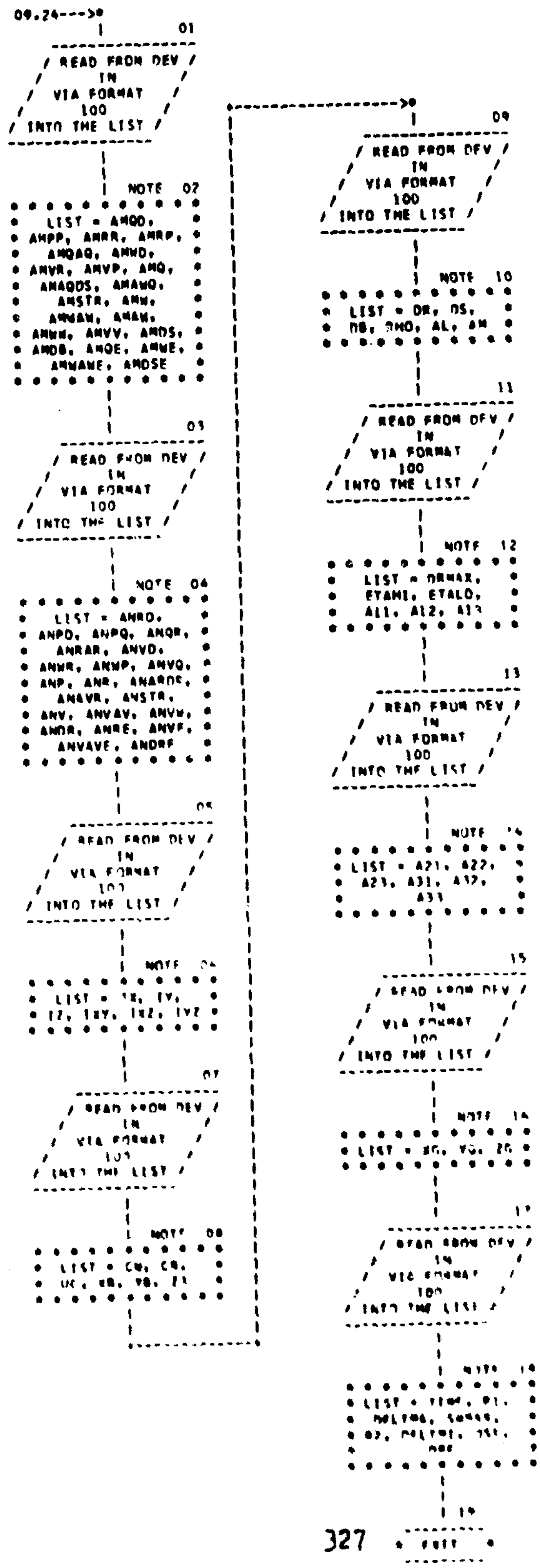
CHART TITLE - SUBROUTINE INPUT



03/24/69

AUTOFLOW CHART SET - EC790 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE INPUT



```
//        JOB    EC780
//        EXEC FFORTRAN
      REAL ICYC
      DIMENSION Y(13) , F(12), F1(12), X(1)
      COMMON XDRDR, XDSDS, XDBDB, A11, A12, A13, A21, A22, A23, A31,
     1        A32, A33, XUD
      COMMON YR, YRD, YPD, YP, YV, YVAV, YDR, YVD
      COMMON ZQ, ZQD, ZRP, ZVP, ZSTP, ZW, ZWAW, ZVV, ZDS, ZDB,
     1        ZWD
      COMMON AKRD, AKP , AKVD, AKV, AKVAV, AKPD
      COMMON AMRP, AMRR, AMWD, AMVR, AMQ, AMAWQ, AMSTR, AMW, AMWAW,
     1        AMVV, AMDS, AMDB, AMQD
      COMMON ANPQ, ANPD, ANVD, ANP, ANR, ANV, ANVAV, ANDR, ANRD
      COMMON DRMAX, ETAHI, ETALO, CW, CB, XG, ZG, AL, AM, DP, DS, DB,
     1    ZB, UC, TIME, R1, DELTMA, SWMAX, R2, DELTMI, DSE, DRE
      COMMON  Y, F, F1, ICYC, NS, H, HH, ISW2
      EQUIVALENCE  (X(1), XDRDR)
      IN = 1
      IOUT = 3
      ICYC = 1.
   20 READ(IN,50) IRUN, NPNT, NS
   50 FORMAT(16I5)
      IF(IRUN)40,30,40
   30 CALL EXIT
   40 CONTINUE
      READ(IN,60) TO, H, TLIM
   60 FORMAT(8F10.5)
      READ(IN,70) (X(I),I=1, 84)
   70 FORMAT(5E13.6)
      ICNT = 1
      Y(13) = TO
      HH = .5*H
      DO 75 I = 1, 12
      F(I) = 0.
   75 F1(I) = 0.
      WRITE(IOUT,77)
      WRITE(IOUT,78) IRUN,NS
      WRITE(IOUT,79)
   77 FORMAT(1H1,25X,20HSUBMARINE SIMULATION/)
   78 FORMAT(1H ,6X,6HRUN NO,I6,5X,3HNS.I4/)
   79 FORMAT(1H ,6X,1HU,14X,1HV,14X,1HW,14X,1HP,14X,1HQ/
     1 1H ,6X,1HR,12X,5HTHETA,11X,3HPSI,13X,3HPHI,13X,1HX/
     2 1H ,6X,1HY,14X,1HZ,14X,1HT/ )
      WRITE(IOUT,80) (Y(I),I=1,13)
   80 FORMAT(1H ,5E15.6)
      WRITE(IOUT,85)
   85 FORMAT(1H )
      ISW2 = 0
   90 CALL UPDATE
      ISW2 = 1
      IF(NPNT-ICNT)100,100,110
  100 ICNT = 0
      WRITE(IOUT,80) (Y(I),I=1,13)
      WRITE(IOUT,85)
```

```
110 ICNT = ICNT + 1
    IF(YLIM-Y(13))20,20,90
    END
```

```
      SUBROUTINE UPDATE
      REAL ICYC
      DIMENSION Y(13) , F(12), F1(12)
      COMMON XDRDR, XDSDS, XDBDB, A11, A12, A13, A21, A22, A23, A31,
     1       A32, A33, XUD
      COMMON YR, YRD, YPD, YP, YV, YVAV, YDR, YVD
      COMMON ZQ, ZQD, ZRR, ZVP, ZSTR, ZW, ZWAW, ZVV, ZDS, ZDB,
     1       ZWD
      COMMON AKRD, AKP , AKVD, AKV, AKVAV, AKPD
      COMMON AMRP, AMRR, AMWD, AMVR, AMQ, AMAWQ, AMSTR, AMW, AMWAW,
     1       AMVV, AMDS, AMDB, AMQD
      COMMON ANPQ, ANPD, ANVD, ANP, ANR, ANV, ANVAV, ANDR, ANRD
      COMMON DRMAX, ETAHI, ETALO, CW, CB, XG, ZG, AL, AM, DR, DS, DB,
     1    ZB, UC, TIME, R1, DELTMA, SWMAX, R2, DELTMI, DSE, DBE
      COMMON  Y, F, F1, ICYC, NS, H, HH, ISW2
      EQUIVALENCE (Y(1),U),(Y(2),V),(Y(3),W),(Y(4),P),(Y(5),Q),(Y(6),R),
     1 (Y(7),THETA),(Y(8),PSI),(Y(9),PHI)
C
C  CALL TO CONTROL
C
      CALL CONTR(THETA)
C
C  COMPUTE QUANTITIES TO BE USED MORE THAN ONCE
C
      U2 = U * U
      V2 = V * V
      RR = R * R
      ROOTVW = SQRT(V2 + W*W)
      VRTVW = V * ROOTVW
      WRTVW = W * ROOTVW
      VR = V * R
      WP = W * P
      RP = R*P
      GWB = ZG*CW - ZB
      WMB = CW - CB
C
C  SET PROPELLER THRUST CONSTANTS
C
      IF(U) 10,20,10
   10 ETA = UC/U
      IF(ETA-ETAHI) 30,20,20
   20 A1 = A11
      A2 = A12
      A3 = A13
      GO TO 60
   30 IF(ETA-ETALO) 40,40,50
   40 A1 = A31
      A2 = A32
      A3 = A33
      GO TO 60
   50 A1 = A21
      A2 = A22
      A3 = A23
   60 CONTINUE
```

```
C
C   COMPUTE TRIG FUNCTIONS
C
        SPHI = SIN(PHI)
        CPHI = COS(PHI)
        STTA = SIN(THETA)
        CTTA = COS(THETA)
        SPSI = SIN(PSI)
        CPSI = COS(PSI)
        TRIG1 = CTTA*SPHI
        TRIG2 = CTTA*CPHI
        TRIG3=SPHI*STTA
        TRIG4 = CPHI * STTA
        TRIG5 = U*CTTA
C
C   COMPUTE UD FROM AXIAL FORCE EQN
C
        F(1) = (AM*(VR-W*Q+XG*(Q*Q+RR)-ZG*(RP+F(5)))+U2*(XDRDR*DR*DR
       1 +XDSDS*DS*DS+XDBDB*DB*DB+A1)+UC*(A2*U+A3*UC)-WMB*STTA)/(AM-XUD)
C
C   COMPUTE VD FROM LATERAL FORCE EQN
C
        F(2) = (U*((YR-AM)*R+YP*P+YV*V)-AM*(ZG*(Q*R-F(4))+XG*(Q*P+F(6)))
       1 +YRD*F(6)+YPD*F(4)+YVAV*VRTVW+YDR*U2*DR+WMB*TRIG1+AM*WP)
       2 /(AM-YVD)
C
C   COMPUTE WD FROM NORMAL FORCE EQN
C
        F(3) = (U*((ZQ+AM)*Q+ZW*W)+AM*(ZG*(P*P+Q*Q)-XG*(RP-F(5)))+
       1 ZQD*F(5)+ZRR*RR+ZVR*VR+ZSTR*U2+ZWAW*WRTVW+ZVV*V2+U2*(ZDS*DS+
       2 ZDB*DB)+WMB*TRIG2) /(AM-ZWD)
C
C   COMPUTE PD FROM ROLLING MOMENT EQN
C
        F(4) = (AM*ZG*(F(2)-WP+U*R)-GWB*TRIG1)*AKPD+AKRD*F(6)+U*(AKP*P
       1 +AKV*V)+AKVAV*VRTVW+AKVD*F(2)
C
C   COMPUTE QD FROM PITCHING MOMENT EQN
C
        F(5) =AMRP*RP-AMQD*AM*(ZG*(F(1)-VR+W*Q)-XG*(F(3)-U*Q+V*P))+AMRR*RR
       1 +AMWD*F(3)+AMVR*VR+U*(AMQ*Q+AMW*W)+U2*(AMSTR+AMDB*DB+AMDS*DS)
       2 +AMAWQ*Q*ROOTVW+AMWAW*WRTVW+AMVV*V2-AMQD*(XG*CW*TRIG2+GWB*STTA)
C
C   COMPUTE RD FROM YAWING MOMENT EQN
C
        F(6) = ANPQ*P*Q-ANRD*AM *(XG*(F(2)-WP+U*R))+ANPD*F(4)+ANVD*F(2)+
       1 U*(ANP*P+ANR*R+ANV*V)+ANVAV*VRTVW+U2*ANDR*DR+ANRD*XG*CW*TRIG1
C
C   COMPUTE KINEMATICS - THETA DOT , PSI DOT, PHI DOT
C
        F(7) = Q*CPHI-R*SPHI
        F(8) =(Q*SPHI+R*CPHI)/CTTA
        F(9) =P+F(8)*STTA
C
```

```
C   COMPUTE  X DOT , Y DOT , Z DOT
C
       F(10)=TRIG5*CPSI+V*(TRIG3*CPSI-CPHI*SPSI)+
     1 W*(TRIG4*CPSI + SPHI*SPSI)
       F(11)=TRIG5*SPSI+V*(TRIG3*SPSI+CPHI*CPSI) +
     1 W*(TRIG4*SPSI-SPHI*CPSI)
       F(12)=-U*STTA+V*TRIG1+W*TRIG2
C
C   INTEGRATE AND REPLACE OLD DERIVATIVES WITH NEW
C
       DO 80 I = 1,12
       Y(I) = Y(I) + HH*(3.*F(I)-F1(I))
       F1(I) = F(I)
    80 CONTINUE
C   UPDATE TIME
       Y(13) = Y(13)+H
       RETURN
       END
```

```
      SUBROUTINE CONTR(THETA)
      REAL ICYC
      DIMENSION Y(13) , F(12), F1(12)
      COMMON XDRDR, XDSDS, XDBDB, A11, A12, A13, A21, A22, A23, A31,
     1       A32, A33, XUD
      COMMON YR, YRD, YPD, YP, YV, YVAV, YDR, YVD
      COMMON ZQ, ZQD, ZRR, ZVP, ZSTR, ZW, ZWAW, ZVV, ZDS, ZDB,
     1       ZWD
      COMMON AKRD, AKP , AKVD, AKV, AKVAV, AKPD
      COMMON AMRP, AMPR, AMWD, AMVP, AMQ, AMAWQ, AMSTR, AMW, AMWAW,
     1       AMVV, AMDS, AMDB, AMQD
      COMMON ANRQ, ANPD, ANVD, ANP, ANR, ANV, ANVAV, ANDR, ANRD
      COMMON DRMAX, ETAHI, ETALO, CW, CB, XG, ZG, AL, AM, DR, DS, DB,
     1    ZB, UC, TIME, R1, DELTMA, SWMAX, R2, DELTMI, DSE, DRE
      COMMON  Y, F, F1, ICYC, NS, H, HH, ISW2
      IF(NS)15,15,16
   15 RETURN
   16 CONTINUE
      GO TO(1001,1001,1003,1004,1005,1006,1007),NS
C
C CONTROL DS
C
 1001 IF(ISW2)21,20,21
   20 N1 = 2
    1 NN2 = 1
      NC2 = ((TIME*ICYC)/H) + .5
      NC3 = (ABS(DS      -DELTMA))*ICYC/ABS(R1*H) +.5
      NC5 = (ABS(DELTMI-DELTMA))*ICYC/ABS(R2*H) +.5
      GO TO 11
   21 GO TO (1,2,3,4,5,11),N1
C
C CYCLES TO START
C
    2 NN2 = NN2 + 1
      IF  (NN2 - NC2) 11,11,7
C
C DS DOWN
C
    7 N1 = 3
      NN3 = 0
    3 NN3 = NN3 +1
      DS = DS  + H*R1/ICYC
      IF(NN3 - NC3) 11,08,8
C
C DS LEVEL
C
    8 N1 = 4
      GO TO 11
    4 IF (ABS(THETA) - SWMAX) 11,9,9
C
C DS UP
C
    9 N1 = 5
      NN5 = 0
```

```
    5 NN5 = NN5 + 1
      DS = DS   + H*R2/ICYC
      IF (NN5 -NC5) 11,10,10
C
C DS LEVEL
C
   10 N1 = 6
C
   11 IF (NS - 2)13,1003,1003
   13 CONTINUE
      GO TO 2000
C
C CONTROL DR + AUTOPILOT
C
 1003 IF (ISW2) 301,300,301
  300 ZC = Y(12)
      SDOT1 = 0.
      DDR = ABS(DRMAX)
      NC10= ((TIME * ICYC)/H) + .5
      NC6 = .85 * DDR * ICYC/(.08726*H) +.5
      NC7 = .08 * DDR * ICYC/(.01336*H) +.5
      NC8 = .04 * DDR * ICYC/(.006*H) +.5
      NC9 = .03 * DDR * ICYC/( .001064*H)+.5
      NN10= 1
      N2 = 2
      IF(DRMAX)313,314,314
  313 R6 = -.08726
      R7 = -.01336
      R8 = -.006
      R9 = -.001064
      GO TO 350
  314 R6 = .08726
      R7 = .01336
      R8 = .006
      R9 = .001064
      GO TO 350
  301 GO TO (300,302,303,304,305,306,350),N2
  302 NN10 = NN10 + 1
      IF (NN10-NC10)350,350,309
C
C FROM 0 TO .85 OF DRMAX
C
  309 N2 = 3
      NN6 = 0
  303 NN6 = NN6 + 1
      DR = DR   + H*R6/ICYC
      IF (NN6 - NC6) 350,309,302
C
C FROM .85 TO .93 OF DRMAX
C
  303 N2 = 4
      NN7 = 0
      GO TO 350
  304 NN7 = NN7 + 1
```

```
      DR = DR   + H*R7/ICYC
      IF (NN7 - NC7) 350,310,310
C
C FROM .93 TO .97 OF DRMAX
C
  310 N2 = 5
      NN8 = 0
      GO TO 350
  305 NN8 = NN8 + 1
      DR = DR   + H*R8/ICYC
      IF (NN8 - NC8) 350,311,311
C
C FROM .97 TO 1. OF DRMAX
C
  311 N2 = 6
      NN9 = 0
      GO TO 350
  306 NN9 = NN9 + 1
      DR = DR   + H*R9/ICYC
      IF (NN9 - NC9) 350,312,312
C
C LEVEL, DRMAX
C
  312 N2 = 7
  350 IF(NS-2)2000,2000,352
C
C AUTOPILOT
C
  352 DSC=.008*(ZC-Y(12))+3.5*Y(7)+.012*(Y(1)*SIN(Y(7))-Y(3)*COS(Y(7)))
     1+2.*Y(5)
  103 IF (DSC) 110,107,107
  107 IF (DSC - .436) 101,108,108
  110 IF (DSC + .436) 109,101,101
  108 DSC = .436
      GO TO 101
  109 DSC = -.436
  101 SDOT = 3 * (DSC -DS)
      DS = DS + .5 * H/ICYC * (3. * SDOT - SDOT1)
      SDOT1 = SDOT
      DB = -DS
  351 CONTINUE
      GO TO 2000
C
C CONTROL DS (IMPULSE), LONGITUDINAL
C
 1024 IF (ISW2)401,400,401
  400 IF(ICYC-1)411,411,412
  411 N4=0
      NTST=1
      NMOD=8
      GO TO 401
  412 N4=-1
      NTST=3
      NMOD=32
```

```
  401 IF(N4-NTST)403,402,403
  402 DS = DSF
  403 IF(MOD(N4,NMOD))410,406,410
C
C PUNCH THETA AND TIME FOR FREQUENCY STUDY (LONGITUDINAL)
C
  406 WRITE(2,408)Y(7),Y(13)
  408 FORMAT(2F15.7)
  410 N4 = N4 + 1
      GO TO 2000
C
C CONTROL DR (IMPULSE), LATERAL
C
 1005 IF (ISW2)501,500,501
  500 IF(ICYC-1)511,511,512
  511 N5 = 0
      NTST =1
      NMOD=8
      GO TO 501
  512 N5=-1
      NTST=3
      NMOD=32
  501 IF(N5-NTST)503,502,503
  502 DR = DRF
  503 IF(MOD(N5,NMOD))510,506,510
C
C PUNCH PHI AND TIME FOR FREQUENCY STUDY(LATERAL)
C
  506 WRITE(2,408) Y(9), Y(13)
  510 N5 = N5 + 1
      GO TO 350
C
C CONTROL ACCEL/DECEL + AUTOPILOT
C
 1006 IF(ISW2)601,600,601
  600 N6=1
      ISW6=0
      NN11=1
      TLIM=     10.*TIME+60.
      ZC = Y(12)
      SDOT1 = 0.
      NC11=60*(ICYC/H)
      NC12=TIME*ICYC/H
      UC=0.
      GO TO 352
  601 GO TO(602,603,604,605,606,607,352),N6
C
      UC=0.
  602 NN11=NN11+1
      IF(NN11-NC11)352,352,608
C
  608 N6=3
      UC=8.445
      NN12=0
```

```
  603 NN12=NN12+1
      IF(NN12-NC12)352,352,609
C
  609 IF(ISW6)617,616,617
  617 N6=7
      UC=0.
      GO TO 352
C
  616 N6=3
      UC=16.89
      NN12=0
  604 NN12=NN12+1
      IF(NN12-NC12)352,352,610
C
  610 IF(ISW6)618,615,618
  618 GO TO 608
C
  615 N6=4
      UC=25.335
      NN12=0
  605 NN12=NN12+1
      IF(NN12-NC12)352,352,611
C
  611 IF(ISW6)619,614,619
  619 GO TO 616
C
  614 N6=5
      UC=33.78
      NN12=0
  606 NN12=NN12+1
      IF(NN12-NC12)352,352,612
C
  612 IF(ISW6)620,621,620
  620 GO TO 615
C
  621 N6=6
      UC=42.225
      NN12=0
  607 NN12=NN12+1
      IF(NN12-NC12)352,352,613
C
  613 ISW6 = 1
      GO TO 614
C
C       CONTROL MAXIMUM ACCEL/DECEL + AUTOPILOT
C
 1007 IF(ISW2)701,700,701
  700 N7=1
      NN13=1
      TLIM=60.+2.*TIME
      NC13=60*ICYC/H
      NC14=TIME*ICYC/H
      SROT1=-1.
      ZC=Y(12)
```

```
      UC=0.
      GO TO 352
  701 GO TO(702,703,352),N7
C
C     UC=0.
  702 NN13=NN13+1
      IF(NN13-NC13)352,352,705
C
  705 N7=2
      UC=42.225
      NN14=0
  703 NN14=NN14+1
      IF(NN14-NC14)352,352,706
C
  706 N7=3
      UC=0.
      GO TO 352
 2000 RETURN
      END
/*
/&
```

AUTOFLOW CHART SET - FC7??

03/24/69

CHART TITLE - PROCEDURES

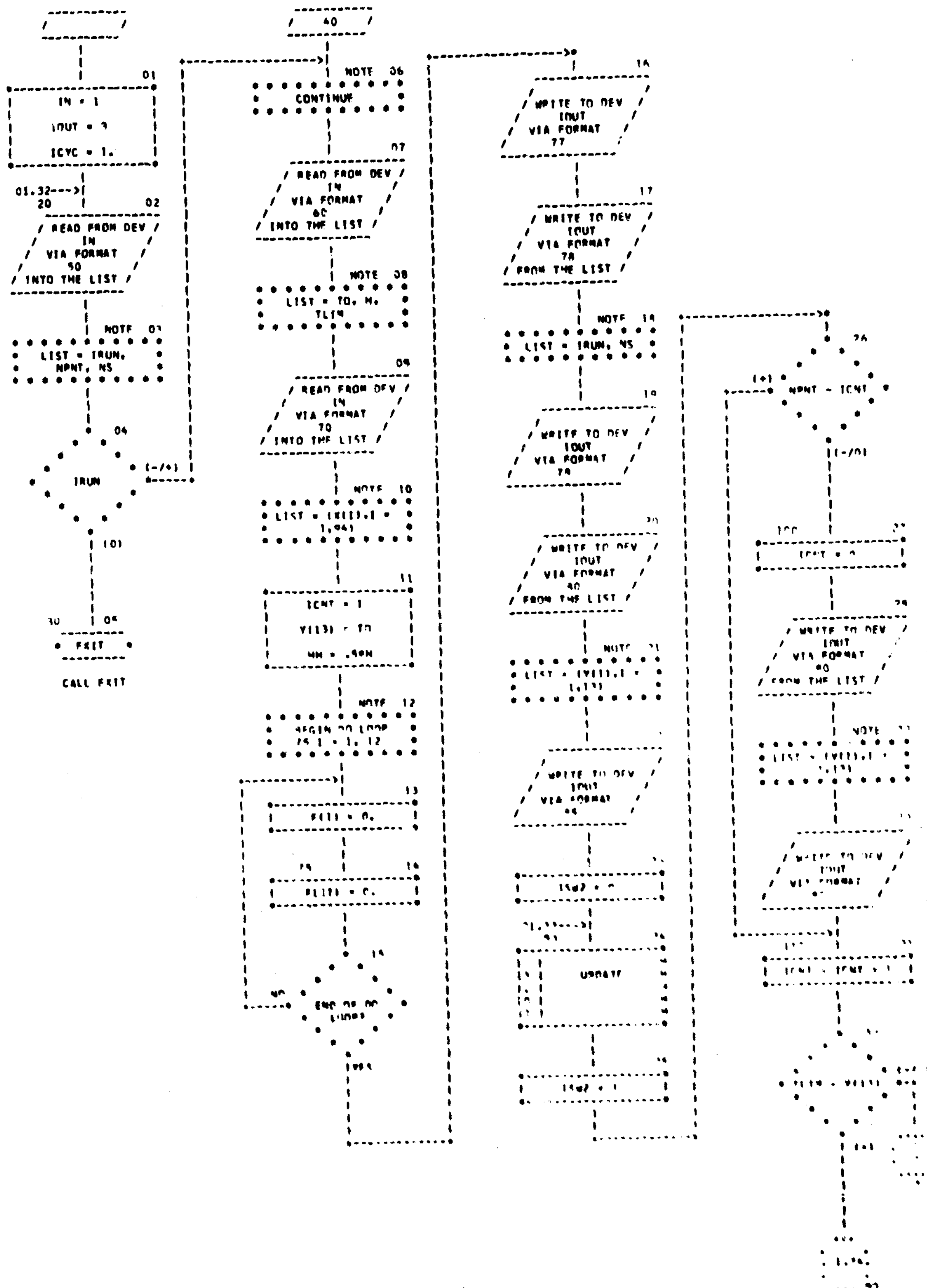


CHART TITLE - SUBROUTINE UPDATE

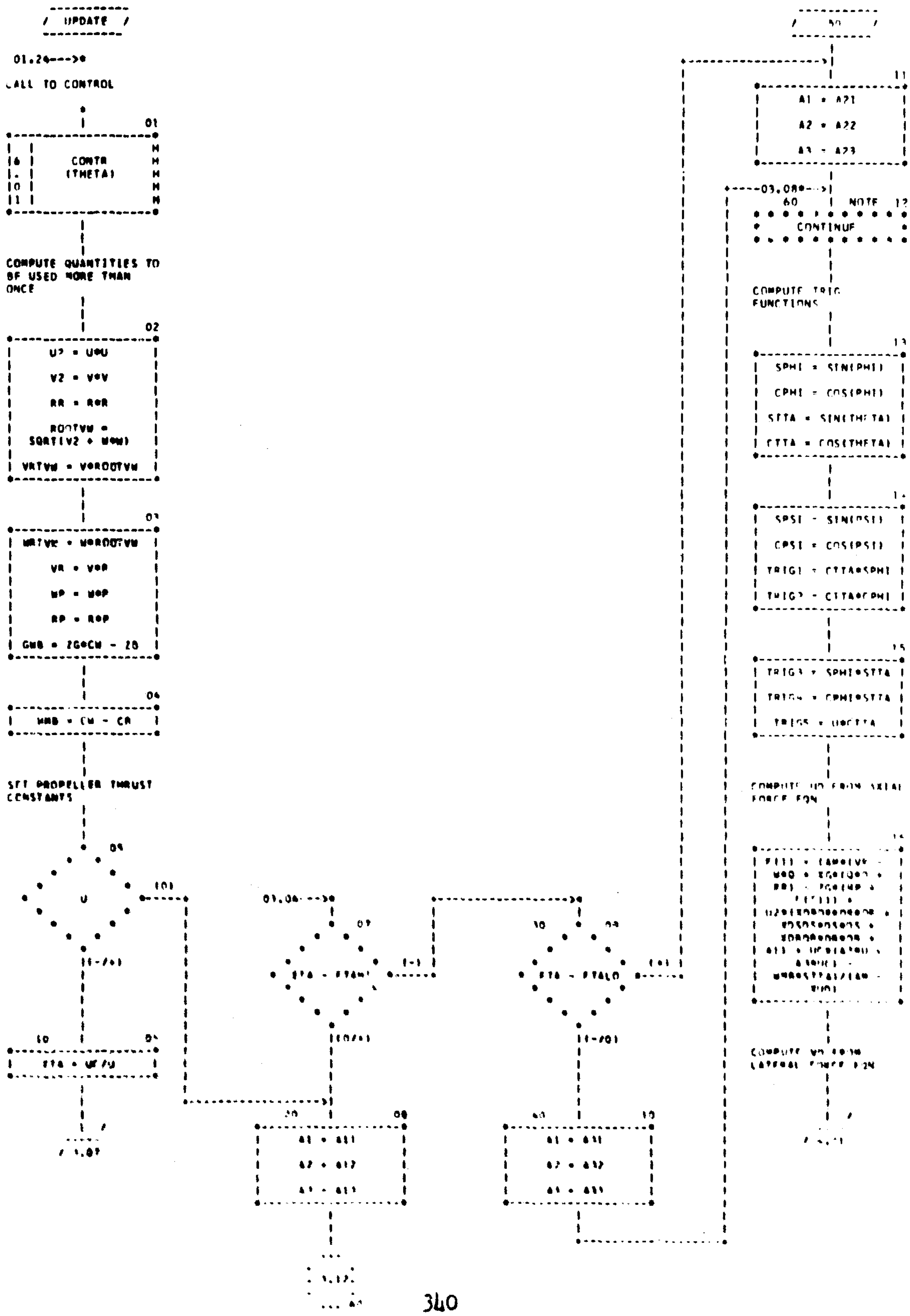
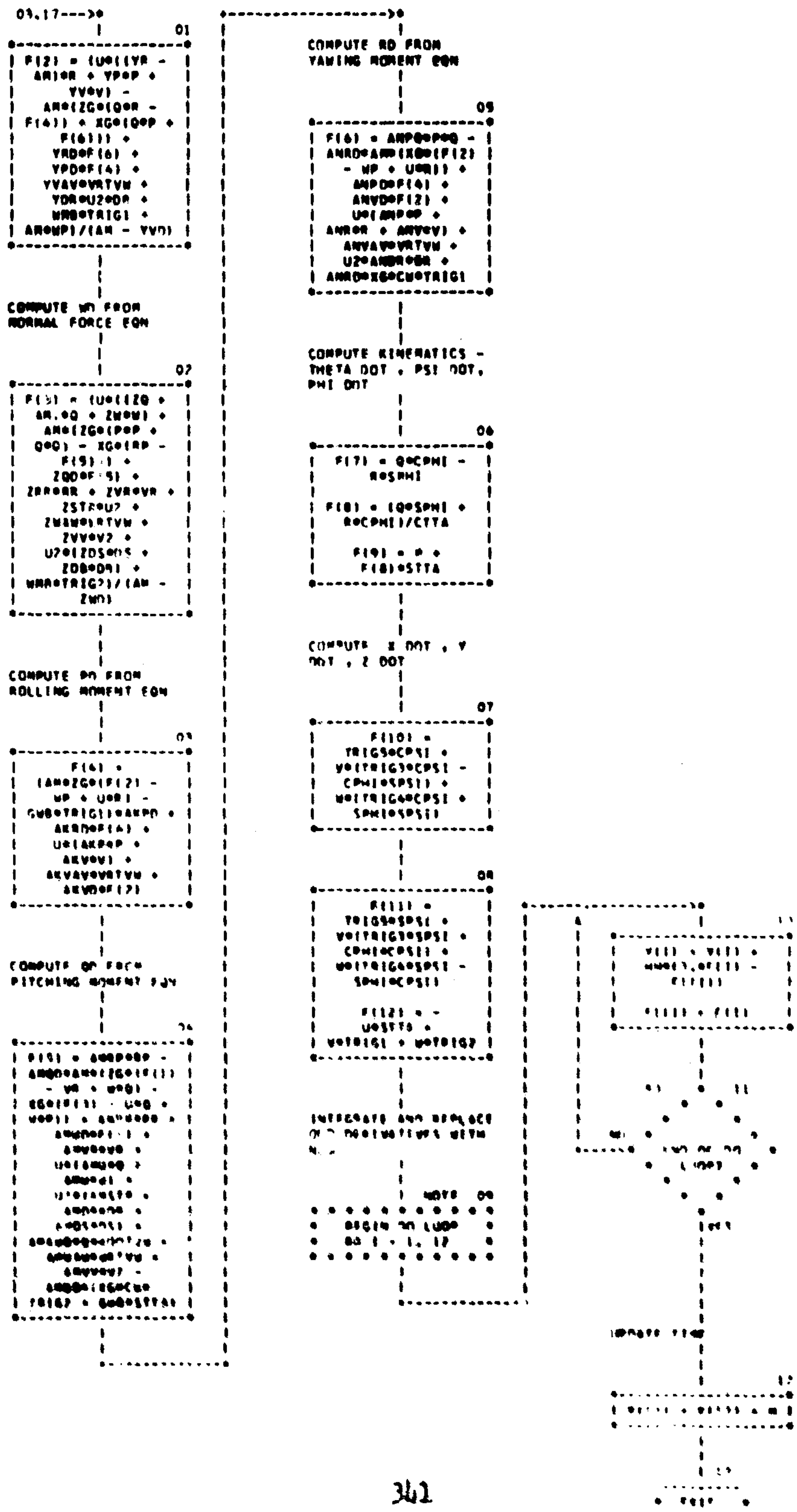
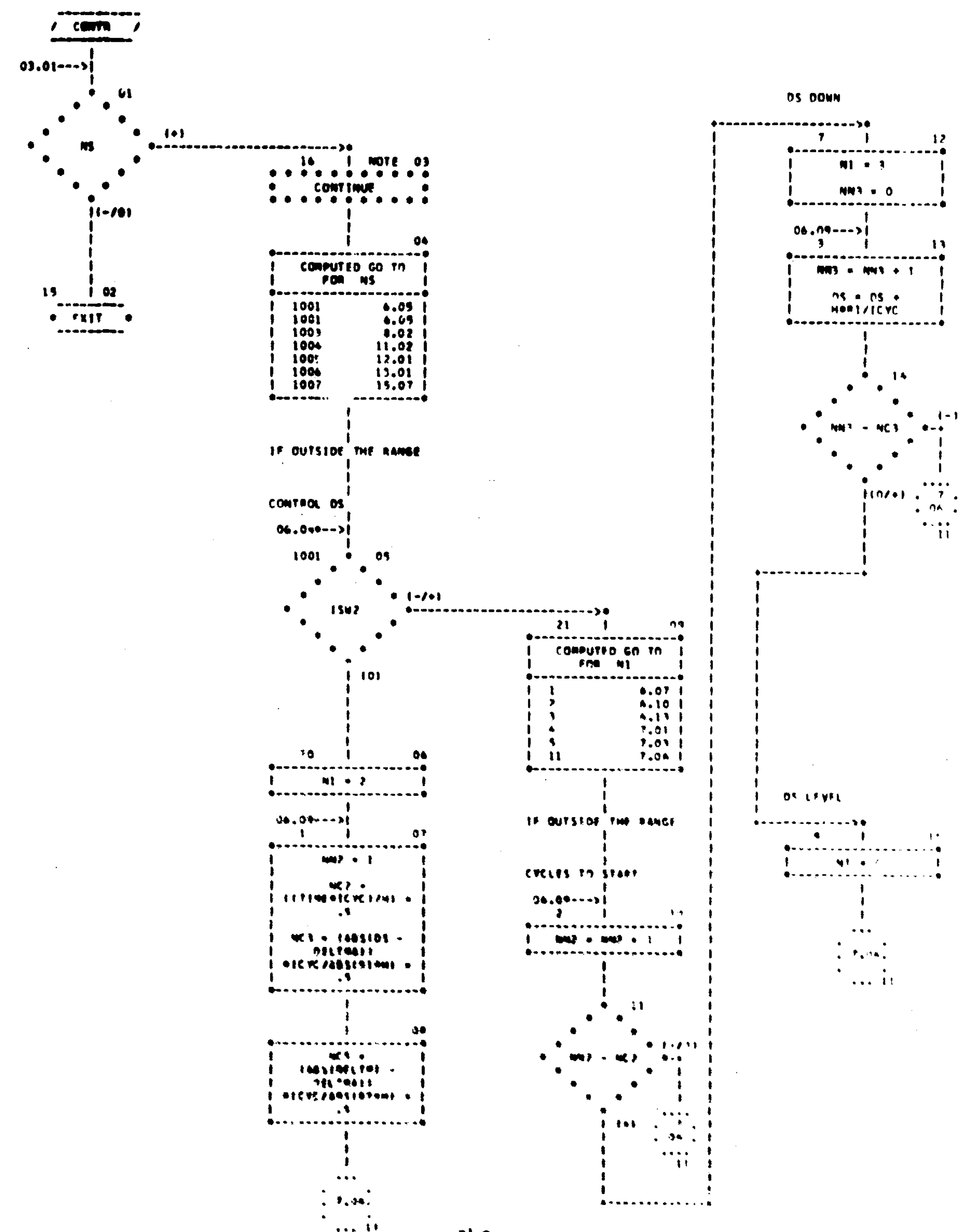


CHART TITLE - SUBROUTINE UPDATE



03/24/69 AUTOFLOW CHART SET - EC780 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE CONTR(THETA)



03/24/69

AUTOFLOW CHART SET - EC780 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE CONTR(THETA)

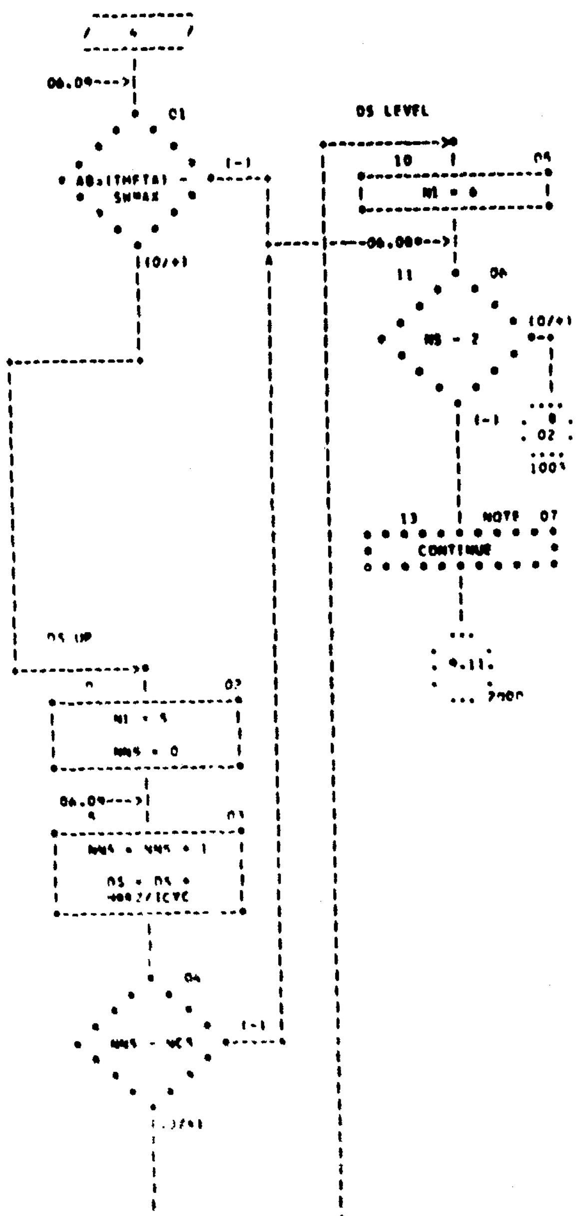


CHART TITLE - SUBROUTINE CONTR(THETA)

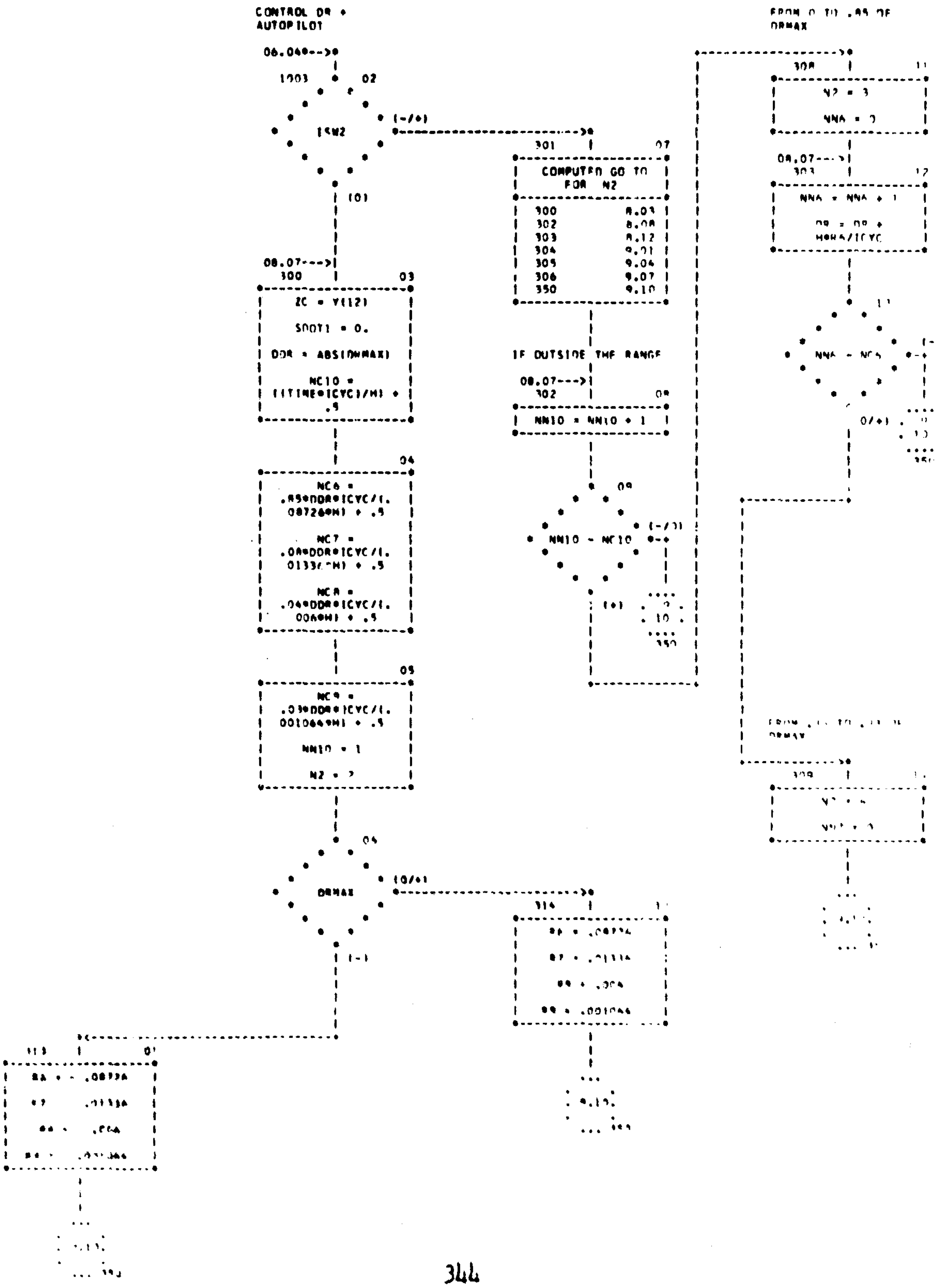


CHART TITLE - SUBROUTINE CONTR(THETA)

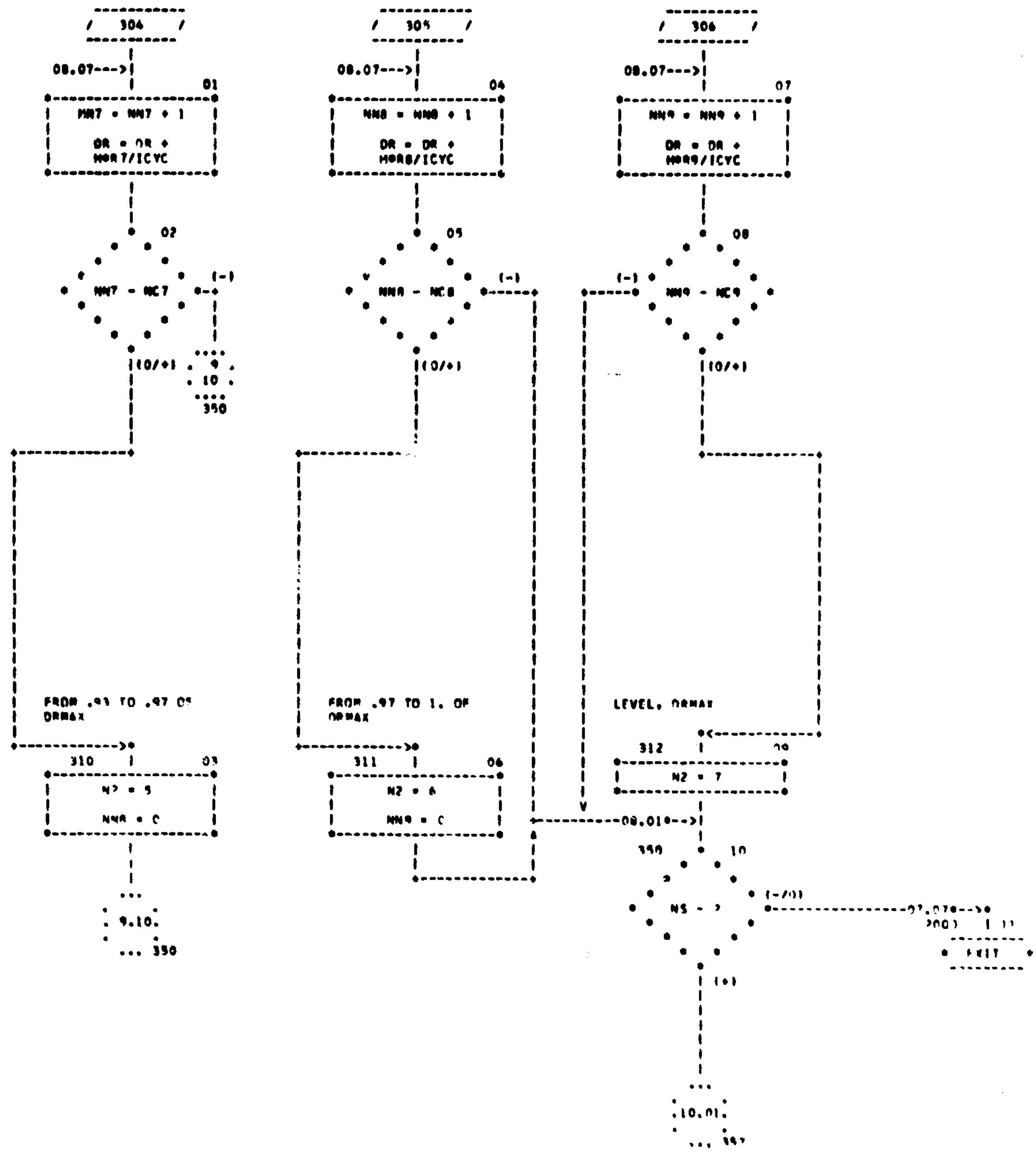
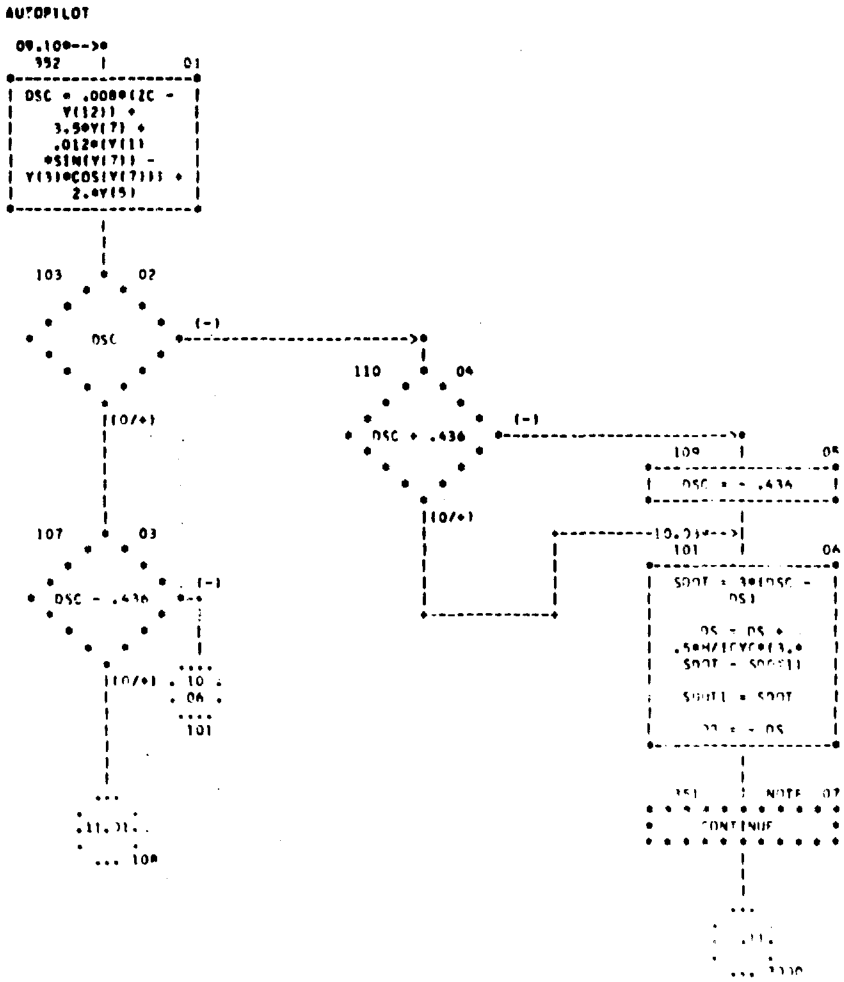
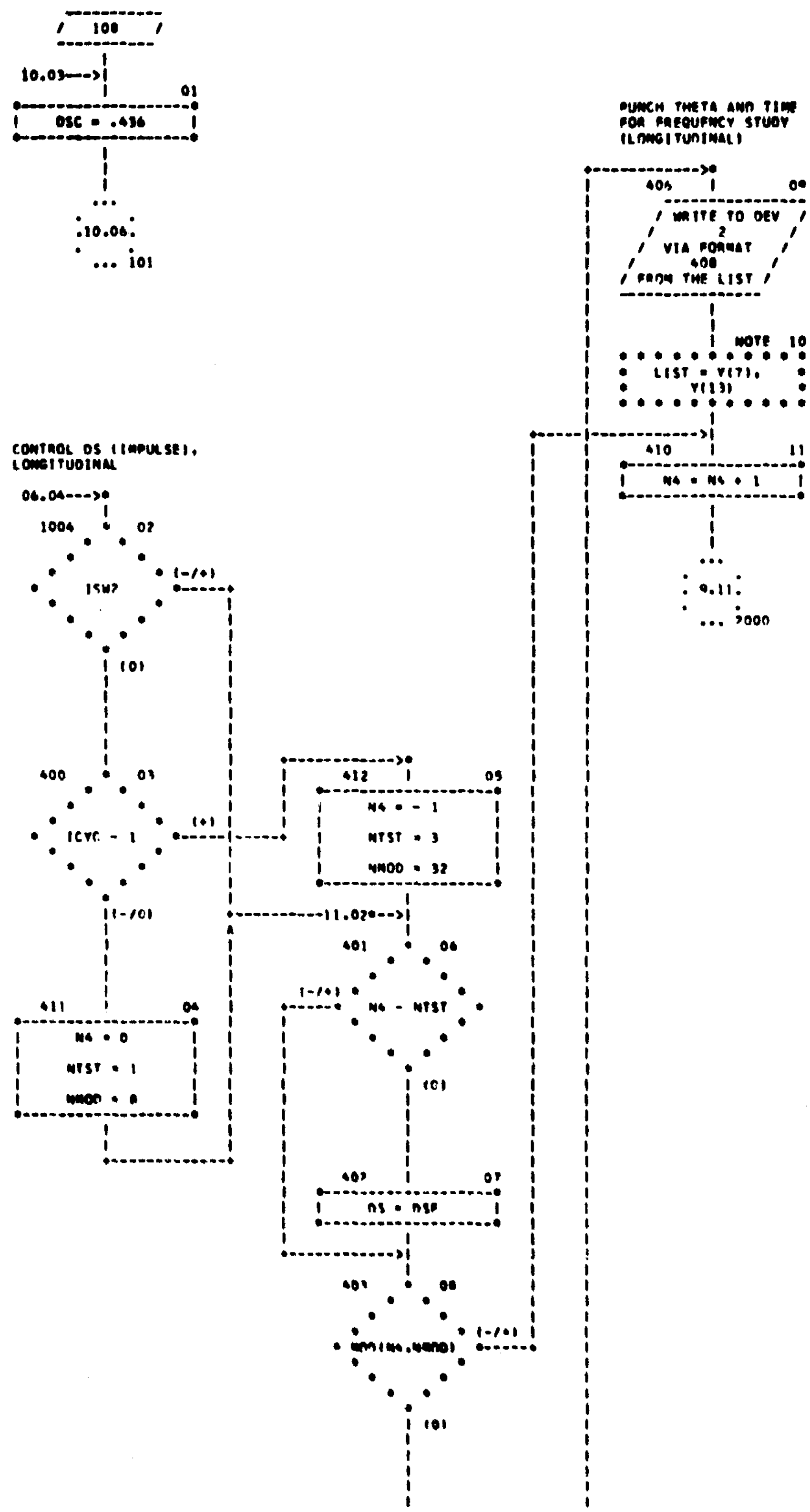


CHART TITLE - SUBROUTINE CONTR(THETA)



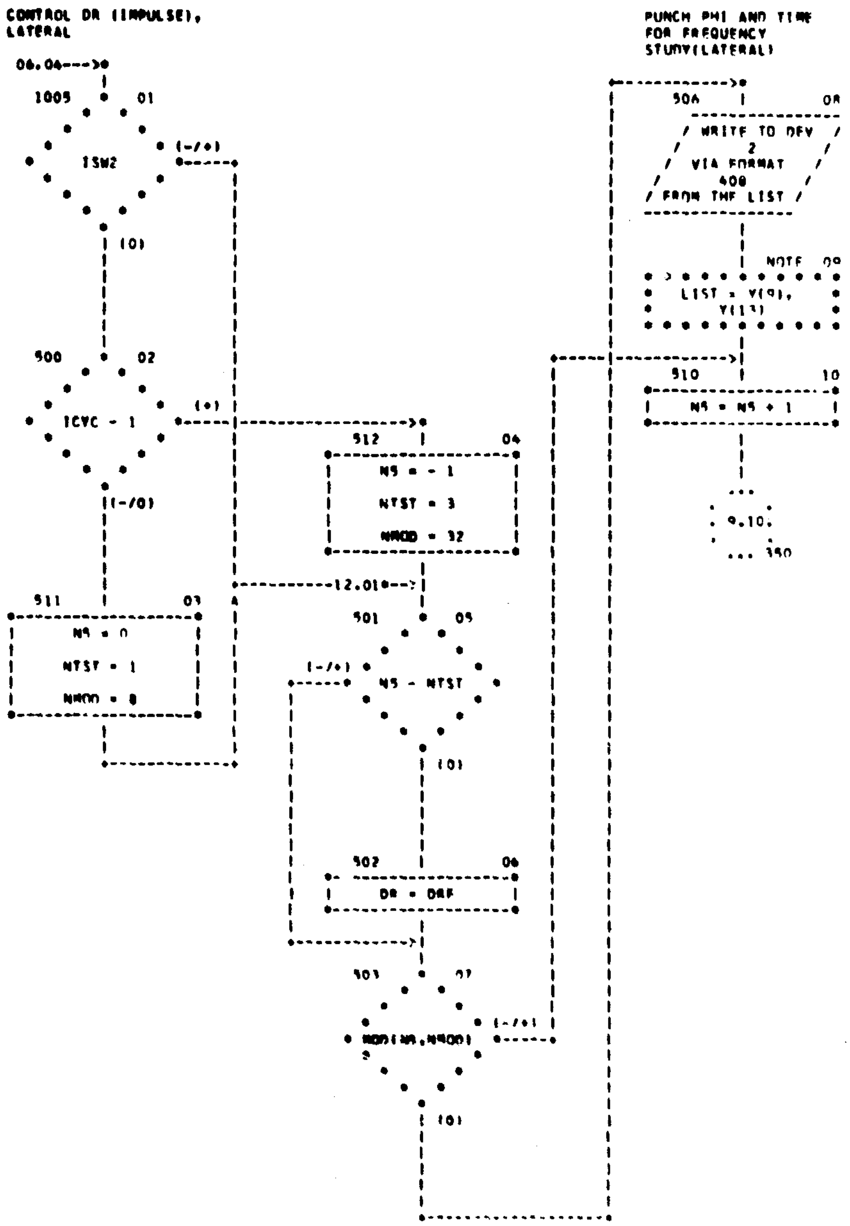
03/24/69 AUTOFLOW CHART SET - EC780 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE CONTR(THETA)



03/24/69 AUTOFLOW CHART SET - EC780 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE CONTR(THETA)



03/24/69 AUTOFLOW CHART SET - SC780 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE CONTR(THETA)

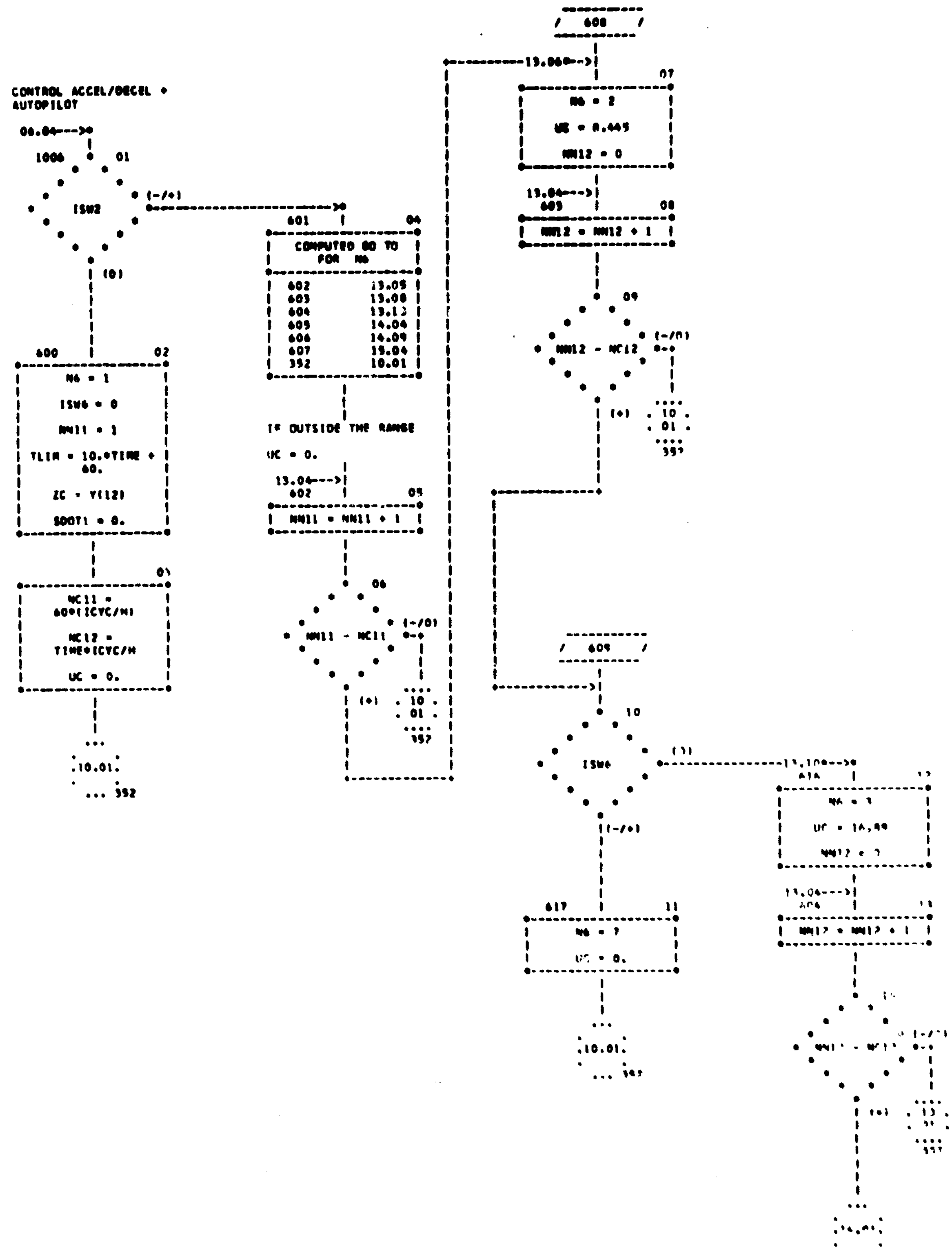
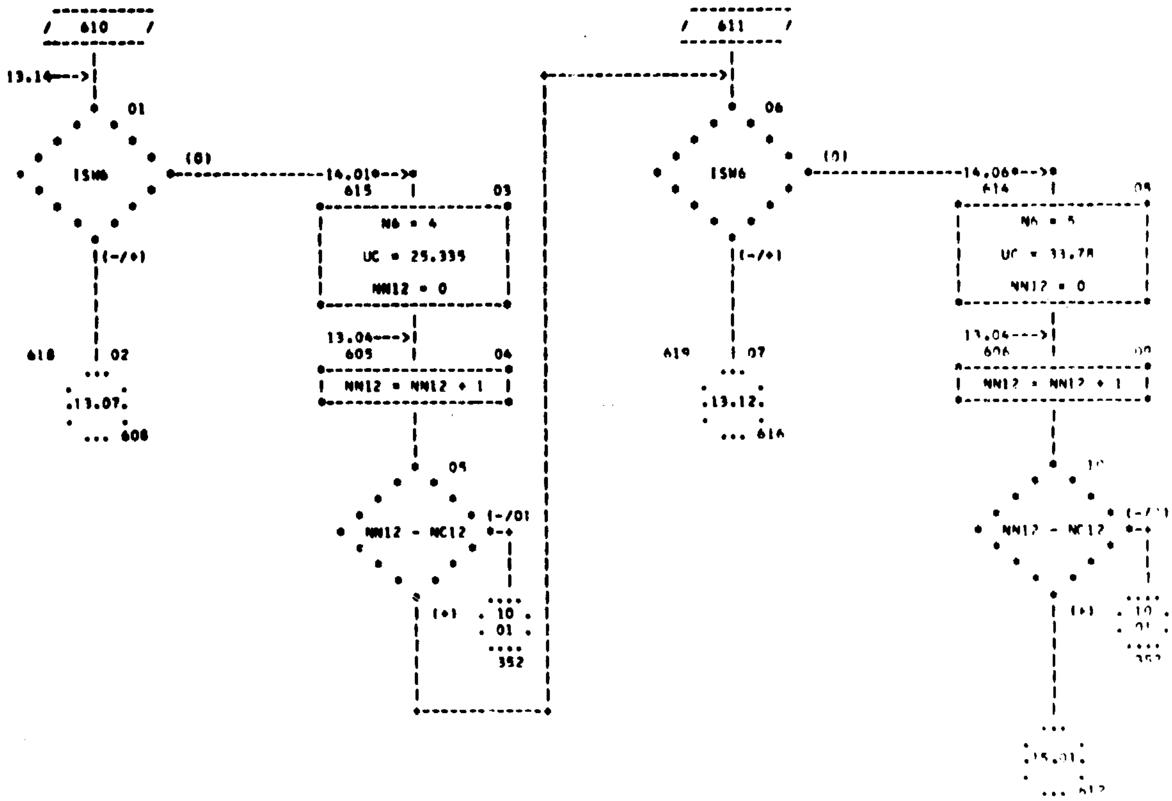
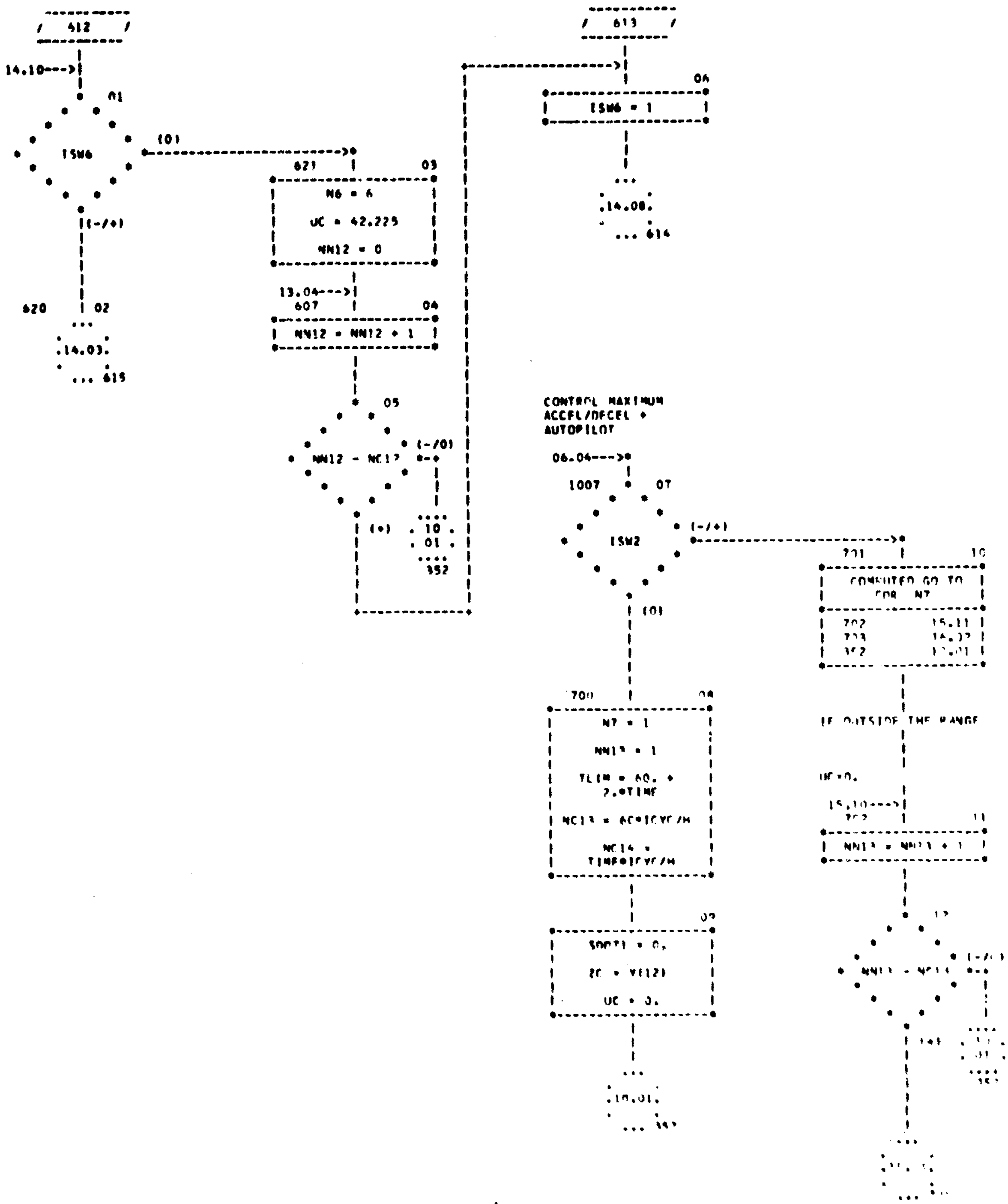


CHART TITLE - SUBROUTINE CONTR(THETA)



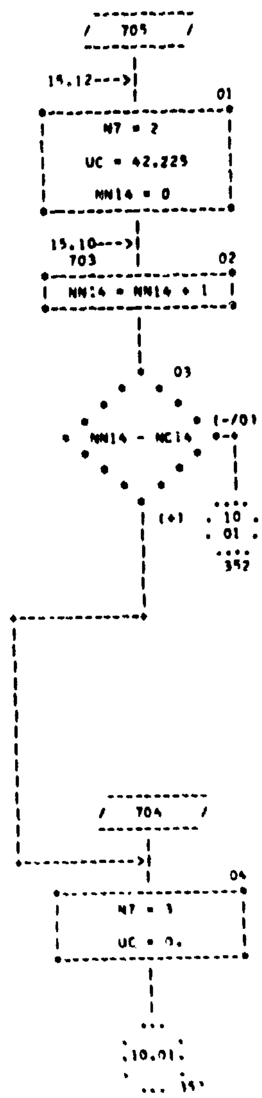
03/24/69 AUTOFLOW CHART SET - FC780 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE CONTR(THETA)



03/24/69 AUTOFLOW CHART SET - EC780 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - SUBROUTINE CONTR(THETA)



```
//        JOB    EC572
//        EXEC FFORTRAN
      DIMENSION VI(8)
      DIMENSION CB(5),AI(5),WJI(10),DWJ(10)
      DIMENSION XP(512),YP(512)
      DIMENSION DATA(2000)
      DIMENSION EPR(50)
    1 FORMAT(8F10.5)
    2 FORMAT(1X,8F15.5)
    3 FORMAT(1X,6E20.7)
    4 FORMAT(4I5,4F10.5)
    5 FORMAT(20A4)
    8 FORMAT(/5X,29HTOO MANY POINTS FOR DIMENSION/)
   11 FORMAT(1H1)
   12 FORMAT(I5,5X,F10.4)
   13 FORMAT(1X,27HPLOT THE TAPE ON UNIT '181')
   14 FORMAT(1X,33HNO PLOT CREATED, SORRY 'BOUT THAT)
   15 FORMAT(/4X,1HN,3X,8HCONTROLS/)
   16 FORMAT(16I5)
   18 FORMAT(/)
   25 FORMAT(17HPSD OF FC572 DATA,10X,3HV =,F10.3,2X,3HKTS
     1       /5F10.4/9X,1H1/9X,1H1/)
   26 FORMAT(3HEND,7X,5H1024.)
      CALL PLOTS(DATA,8000,5)
      IIN = 1
      IOUT=3
      IPCH = 2
      ICNSL = 15
      NDP = 512
      IUSED = 0
      ZERC = 0.
      XPSD = 8.
      YPSD = 8.
      IHX = IHEX(14,3,4,0,4,0,4,0)
      IHY = IHEX(14,8,4,0,4,0,4,0)
C
      PI = 3.141592654
      CALL INIT(-1)
      TWOPI = 6.283185308
      SQRT3D = .8660254
      LM = 5
      CB(1) = .5
      CB(2) = SQRT3D
      CB(3) = 1.
      CB(4) = SQRT3D
      CB(5) = .5
      AI(1) = .09058
      AI(2) = .43305
      AI(3) = .53254
      AI(4) = .43305
      AI(5) = .09058
C
      G=32.2
      CONG = 9.10E-3 *G*G
```

```
      BC= (CONG+CONG)/PI
      READ(IIN,4) NBAND,NT,NPLT,NPCH,A,B,DT
      WRITE(IOUT,15)
      NBAND = 10
      IF(NPLT)115,115,112
  112 IF( NT - NPLT*NDP )115,115,113
  113 CONTINUE
      NPLT = (NT-10)/NDP + 1
  115 CONTINUE
      WRITE(IOUT,4) NBAND,NT,NPLT,NPCH,A,B,DT
      NPLTS = NPLT
      BAND = NBAND
  100 CONTINUE
      READ(IIN,1) VI
C     TEST FOR EOF
      IF(VI(1))999,999,105
  105 CONTINUE
      DO 600 IV=1,8
      V = VI(IV)
      IF(V)605,605,205
  205 CONTINUE
      WRITE(IOUT,11)
      U = V*1.688944
      GOU = G/U
      WRITE(IOUT,2) V,U,GOU
      FACT1= (8.1E-3*U*U)/(2.96*GOU*GOU)
      FACT2= -.74*GOU*GOU*GOU*GOU
      YO = EXP(FACT2/(A*A*A*A))
      AY = EXP(FACT2/(B*B*B*B)) - YO
      AREA = FACT1*AY
      AR = AREA/BAND
      WRITE(IOUT,3) FACT1,FACT2,YO,AY,AREA,AR
      AY = AY/BAND
      WJ = A
      IJ = 0
      DO 200 J=1,NBAND
      YO = YO + AY
      AT = -.74/ALOG(YO)
      X1 = GOU*SQRT(SQRT(AT))
      Y1 = YO*FACT1
      WRITE(IOUT,3) X1,Y1,YO
      WJM1 = WJ
      WJ = X1
      WJI(J) = .5*(WJ+WJM1)
      DWJ(J) = WJ - WJM1
      DO 200 L=1,LM
      IJ = IJ + 1
      CALL RANDM(RN)
      EPR(IJ) = RN*TWOPI
  200 CONTINUE
C
      JJ = 0
      DO 500 IT=1,NT
      TSUM = 0.
```

```
      TT= FLOAT(IT-1)*DT
      IJ = 0
      DO 400 J=1,NBAND
      WJ = WJI(J)
      DW = DWJ(J)
      WJ2 = WJ*WJ
      WJ4 = WJ2*WJ2
      WJ5 = WJ4*WJ
      AR = BC*EXP(FACT2/WJ4)/WJ5
      AT = WJ2/GOU
      SUM = 0.
      DO 390 L=1,LM
      IJ = IJ + 1
      EIJ = EPR(IJ)
      GIJ = (AT*CB(L) - WJ)*TT + EIJ
      SUM = SUM + COS(GIJ)*SQRT(AI(L)*DW)
  390 CONTINUE
      TSUM = TSUM + AR*SUM
  400 CONTINUE
C     WRITE(IOUT,3)TT,TSUM
C
      IF(NPLT)425,425,405
  405 IF((IT/NPLT)*NPLT-IT)425,410,425
  410 JJ = JJ+1
      IF(JJ-NDP)420,420,415
  415 WRITE(IOUT,8)
      IF(NPCH)525,525,418
  418 NPLT= 0
      GO TO 425
  420 CONTINUE
      XP(JJ)=TT
      YP(JJ)=TSUM
  425 CONTINUE
C
      IF(NPCH)465,465,450
  450 CONTINUE
      IF(IT-1)452,452,453
  452 WRITE(IPCH,25) V,ZERO,DT,XPSD,YPSD,ZERO,ZERO
  453 CONTINUE
      WRITE(IPCH,12) IT,TSUM
      IF(IT-NT)465,460,460
  460 WRITE(IPCH,26)
  465 CONTINUE
C
  500 CONTINUE
C     PLOT
  525 CONTINUE
      NPLT = NPLTS
      IF(NPLT)575,575,550
  550 CONTINUE
      J = MIN0(JJ,NDP)
      XL = 9.
      YL = 10.
      DIV = 10.
```

```
      HT = .125
      CALL SCALE(XP,XL,J,1,DIV,1)
      CALL SCALE(YP,YL,J,1,DIV,2)
      CALL AXIS(ZERO,ZERO,IHX,-4,XL,0.0,DIV,1)
      CALL AXIS(ZERO,ZERO,IHY, 4,YL,90.,DIV,2)
      CALL LINE(XP,YP,J,1,0,0)
      IUSED = IUSED + 1
      CALL PLOT(XL+4.,0.,-3)
  575 CONTINUE
C
  600 CONTINUE
  605 CONTINUE
      GO TO 100
  999 CONTINUE
      IF(IUSED)998,997,998
  998 CONTINUE
      CALL PLOT(8.,0.,999)
      WRITE(ICNSL,13)
      GO TO 996
  997 CONTINUE
      WRITE(ICNSL,14)
  996 CONTINUE
      CALL EXIT
      END
/*
/&
```

CHART TITLE - PROCEDURES

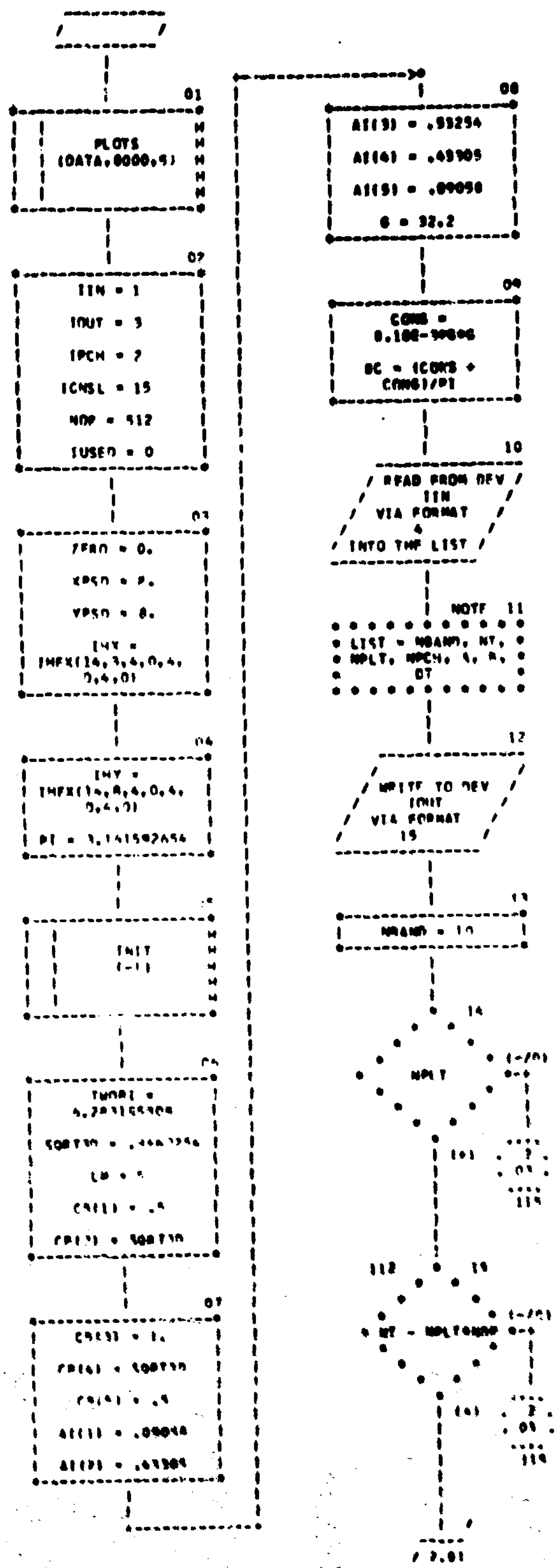
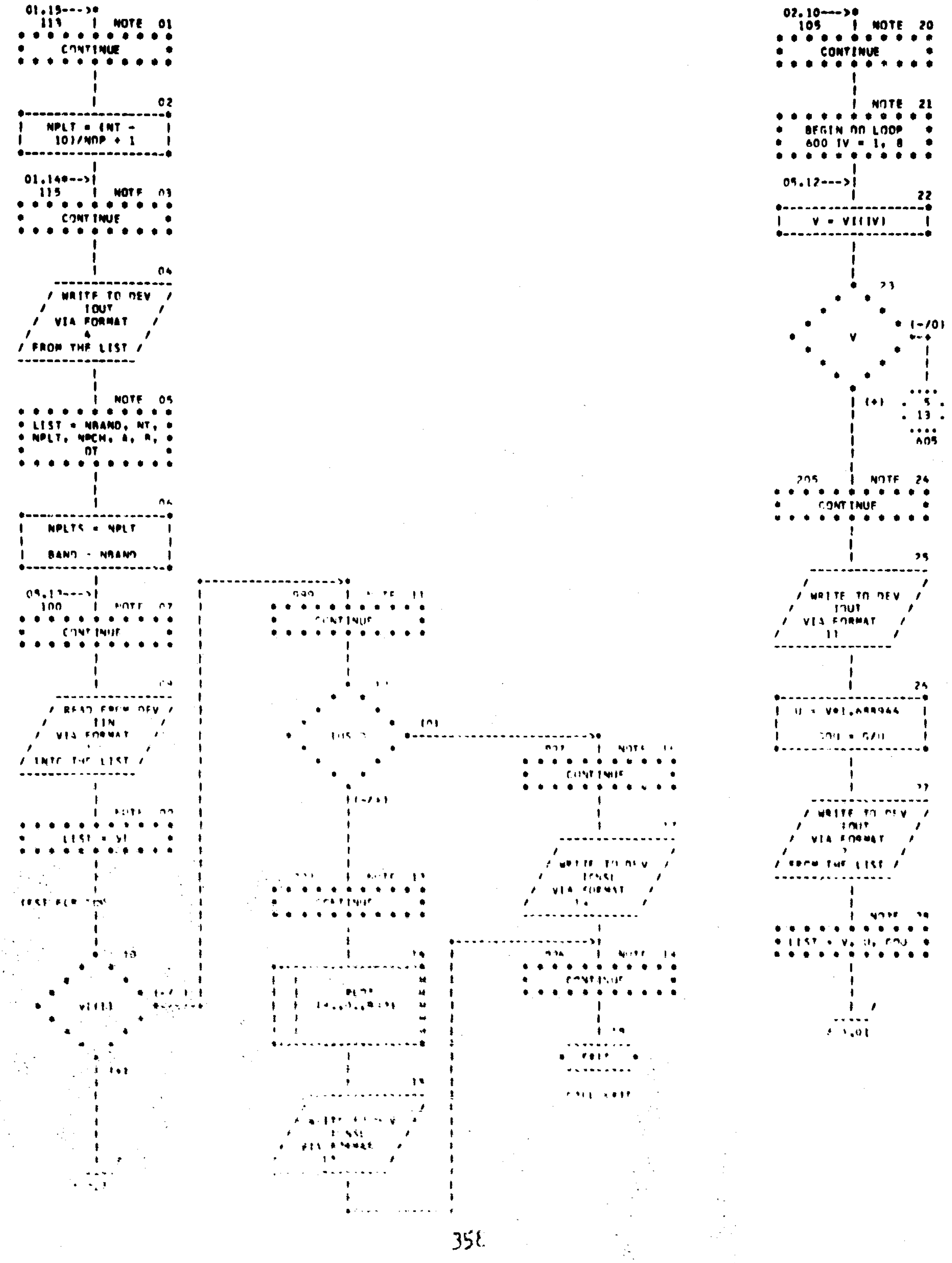
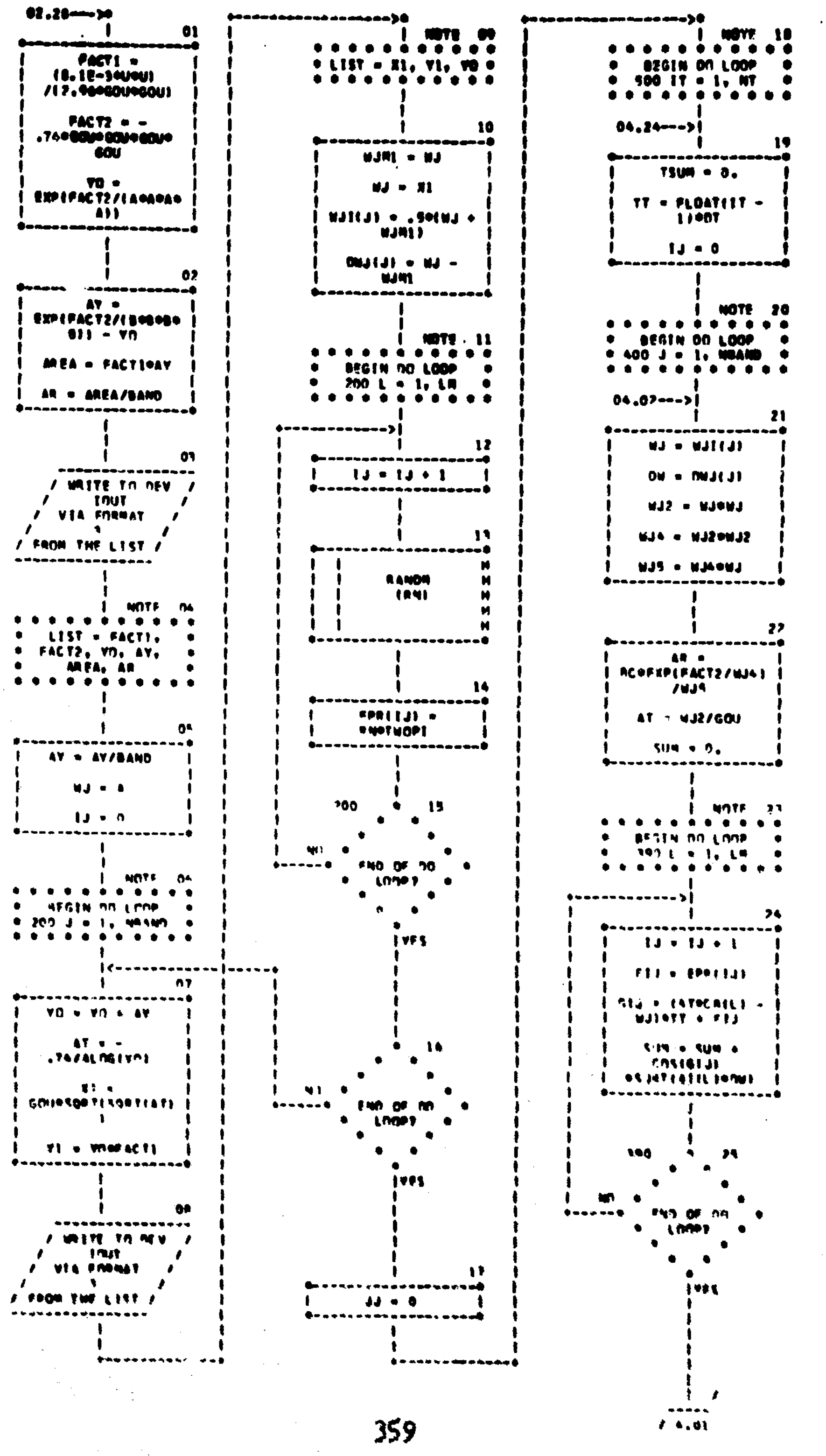


CHART TITLE - PROCEDURES




02.20--->
01
FACT1 = (8.1E-5*U*U) /(2.96*GOU*GOU)
FACT2 = -.74*GOU*GOU*GOU*GOU
VO = EXP(FACT2/(A*A*A*A))
02
AV = EXP(FACT2/(B*B*B*B)) - VO
AREA = FACT1*AV
AR = AREA/BAND
03
WRITE TO DEV IOUT VIA FORMAT FROM THE LIST
NOTE 04
LIST = FACT1, FACT2, VO, AV, AREA, AR
05
AV = AV/BAND
WJ = A
IJ = 0
NOTE 06
BEGIN DO LOOP 200 J = 1, NBAND
07
VO = VO + AV
AT = -.74/ALOG(VO)
X1 = GOU*SQRT(SQRT(AT))
Y1 = VO*FACT1
08
WRITE TO DEV IOUT VIA FORMAT FROM THE LIST
NOTE 09
LIST = X1, Y1, VO
10
WJM1 = WJ
WJ = X1
WJI(J) = .5*(WJ + WJM1)
DWJ(J) = WJ - WJM1
NOTE 11
BEGIN DO LOOP 200 L = 1, LN
12
IJ = IJ + 1
13
RANDM (RN)
14
EPR(IJ) = RN*TWOPI
200
15
NO
END OF DO LOOP?
YES
16
NO
END OF DO LOOP?
YES
17
IJ = 0
NOTE 18
BEGIN DO LOOP 500 IT = 1, NT
04.24--->
19
TSUM = 0.
TT = FLOAT(IT - 1)*DT
IJ = 0
NOTE 20
BEGIN DO LOOP 400 J = 1, NBAND
04.07--->
21
WJ = WJI(J)
DW = DWJ(J)
WJ2 = WJ*WJ
WJ4 = WJ2*WJ2
WJ5 = WJ4*WJ
22
AR = RC*EXP(FACT2/WJ4) /WJ5
AT = WJ2/GOU
SUM = 0.
NOTE 23
BEGIN DO LOOP 300 L = 1, LN
24
IJ = IJ + 1
FII = EPR(IJ)
SIJ = (ATCR(L) - WJ)*TT + FIJ
SUM = SUM + COS(SIJ)
300
25
NO
END OF DO LOOP?
YES
4.01

03/11/69 AUTOFLOW CHART SET - EC572 NAVTRADEVCEN 68-C-0050-2

CHART TITLE - PROCEDURES

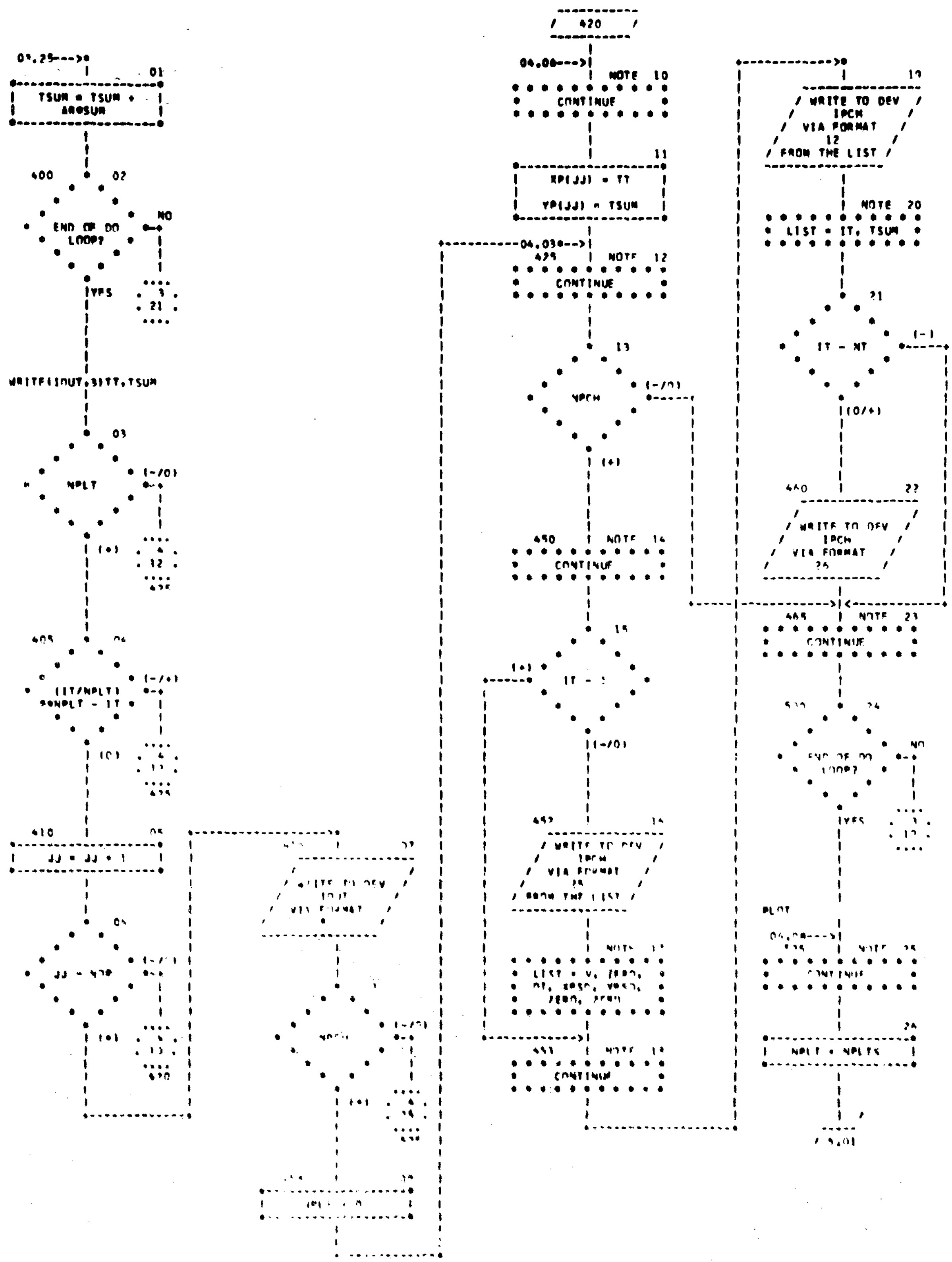
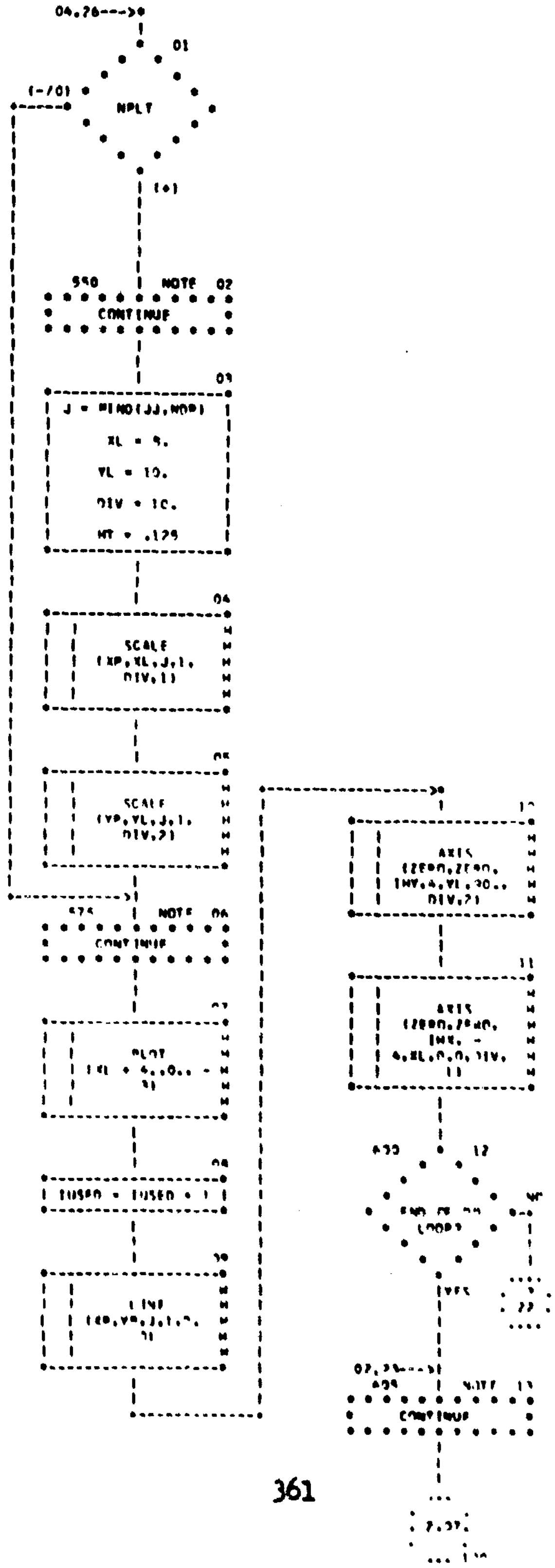


CHART TITLE - PROCEDURES



APPENDIX B

INTEGRATION TECHNIQUES

The subroutine INTEG used with program EB920 Submarine Simulation is programed to use three different integration techniques; Euler, 2nd Order Adams, and a 2nd non-classical method (O_{12}). Table 24 contains the coefficients for these and 19 other methods that can be programed into this subroutine by means of the equation

$$Y_n = \sum_{i=1}^{3} a_i Y_{n-i} + h \sum_{i=0}^{3} b_i \dot{Y}_{n-i}$$

TABLE 25, POPULAR NUMERICAL INTEGRATION TECHNIQUES

$$Y_n = \sum_{i=1}^{3} a_i Y_{n-i} + h \sum_{i=0}^{3} b_i \dot{Y}_{n-i}$$

Method	Type	a_1	a_2	a_3	b_0	b_1	b_2	b_3
Euler	O_{11}	1				1		
Backward Rectangular	C_{11}	1			1			
2nd Order Adams	O_{12}	1				3/2	-1/2	
Trapezoidal	C_{12}	1			1/2	1/2		
O_{33} Mod Gurk †	O_{33}	1.1462	-0.2011	0.0549		1.6416	-1.0080	0.2751
Classic O_{33}	O_{33}	-18	9	10		9	18	3
Simpson	C_{13}		1		1/3	4/3	1/3	
O_{30} C_{31} Mod Gurk †	O_{30}	1.807	-1.109	0.303				
	C_{31}	1.146	-0.201	0.055	0.909			
Classic O_{30}	O_{30}	3	-3	1				
Classic C_{31}	C_{31}	18/11	-9/11	2/11	6/11			
3/8 Rule	C_{14}			1	3/8	9/8	9/8	3/8
Adams - Bashforth	C_{14}	1			9/24	19/24	-5/24	1/24
Best O_{12} Method Based † on Stability Alone	O_{12}	1				3/4	1/4	
1/2 Rule	C_{24}	1/2	1/2		17/48	51/48	3/48	1/48
Parabolic	O_{13}	1				23/12	-4/3	5/12
Classic	O_{11}		1			2		
Classic	O_{22}	-4	5			4	2	
Classic	C_{22}	8/10	2/10		4/10	8/10		
Classic	C_{13}	1			5/12	2/3	-1/12	
Classic	C_{32}	9/17	9/17	-1/17	6/17	18/17		
1/3 Rule	C_{34}	1/3	1/3	1/3	13/36	39/36	15/36	5/36
2/3 Rule	C_{24}		2/3	1/3	25/72	91/72	43/72	9/72

† Denotes a non-classic method

GLOSSARY

Symbol	Dimensionless Form	Definition
B	$B' = \frac{B}{\frac{1}{2}\rho \ell^2 U^2}$	Buoyancy force, positive upward
CB		Center of buoyancy of submarine
CG		Center of mass of submarine
I_x	$I_x' = \frac{I_x}{\frac{1}{2}\rho \ell^5}$	Moment of inertia of submarine about x axis
I_y	$I_y' = \frac{I_y}{\frac{1}{2}\rho \ell^5}$	Moment of inertia of submarine about y axis
I_z	$I_z' = \frac{I_z}{\frac{1}{2}\rho \ell^5}$	Moment of inertia of submarine about z axis
I_{xy}	$I_{xy}' = \frac{I_{xy}}{\frac{1}{2}\rho \ell^5}$	Product of inertia about xy axis
I_{yz}	$I_{yz}' = \frac{I_{yz}}{\frac{1}{2}\rho \ell^5}$	Product of inertia about yz axes
I_{zx}	$I_{zx}' = \frac{I_{zx}}{\frac{1}{2}\rho \ell^5}$	Product of inertia about zx axes
K	$K' = \frac{K}{\frac{1}{2}\rho \ell^3 U^2}$	Hydrodynamic moment component about x axis (rolling moment)
K_*	$K_*' = \frac{K_*}{\frac{1}{2}\rho \ell^3 U^2}$	Rolling moment when body angle (α, β) and control surface angles are zero
$K_{*\eta}$	$K_{*\eta}' = \frac{K_{*\eta}}{\frac{1}{2}\rho \ell^3 U^2}$	Coefficient used in representing K_* as a function of $(\eta - 1)$
K_p	$K_p' = \frac{K_p}{\frac{1}{2}\rho \ell^4 U}$	First order coefficient used in representing K as a function of p
$K_{\dot{p}}$	$K_{\dot{p}}' = \frac{K_{\dot{p}}}{\frac{1}{2}\rho \ell^5}$	Coefficient used in representing K as a function of $\dot{p}$
$K_{p\|p\|}$	$K_{p\|p\|}' = \frac{K_{p\|p\|}}{\frac{1}{2}\rho \ell^5}$	Second order coefficient used in representing K as a function of p
K_{pq}	$K_{pq}' = \frac{K_{pq}}{\frac{1}{2}\rho \ell^5}$	Coefficient used in representing K as a function of the product pq

K_{qr}	$K_{qr}' = \frac{K_{qr}}{\frac{1}{2}\rho \ell^5}$	Coefficient used in representing K as a function of the product qr
K_r	$K_r' = \frac{K_r}{\frac{1}{2}\rho \ell^4 U}$	First order coefficient used in representing K as a function of r
$K_{\dot{r}}$	$K_{\dot{r}}' = \frac{K_{\dot{r}}}{\frac{1}{2}\rho \ell^5}$	Coefficient used in representing K as a function of $\dot{r}$
K_v	$K_v' = \frac{K_v}{\frac{1}{2}\rho \ell^3 U}$	First order coefficient used in representing K as a function of v
$K_{\dot{v}}$	$K_{\dot{v}}' = \frac{K_{\dot{v}}}{\frac{1}{2}\rho \ell^4}$	Coefficient used in representing K as a function of $\dot{v}$
$K_{v\|v\|}$	$K_{v\|v\|}' = \frac{K_{v\|v\|}}{\frac{1}{2}\rho \ell^3}$	Second order coefficient used in representing K as a function of v
K_{vq}	$K_{vq}' = \frac{K_{vq}}{\frac{1}{2}\rho \ell^4}$	Coefficient used in representing K as a function of the product vq
K_{vw}	$K_{vw}' = \frac{K_{vw}}{\frac{1}{2}\rho \ell^3}$	Coefficient used in representing K as a function of the product vw
K_{wp}	$K_{wp}' = \frac{K_{wp}}{\frac{1}{2}\rho \ell^4}$	Coefficient used in representing K as a function of the product wp
K_{wr}	$K_{wr}' = \frac{K_{wr}}{\frac{1}{2}\rho \ell^4}$	Coefficient used in representing K as a function of the product wr
$K_{\delta r}$	$K_{\delta r}' = \frac{K_{\delta r}}{\frac{1}{2}\rho \ell^3 U^2}$	First order coefficient used in representing K as a function of δ_r
ℓ	$\ell' = 1$	Overall length of submarine
m	$m' = \frac{m}{\frac{1}{2}\rho \ell^3}$	Mass of submarine, including water in free-flooding spaces
M	$M' = \frac{M}{\frac{1}{2}\rho \ell^3 U^2}$	Hydrodynamic moment component about y axis (pitching moment)
M_*	$M_*' = \frac{M_*}{\frac{1}{2}\rho \ell^3 U^2}$	Pitching moment when body angles (α, θ) and control surface angles are zero
M_{pp}	$M_{pp}' = \frac{M_{pp}}{\frac{1}{2}\rho \ell^5}$	Second order coefficient used in representing M as a function of p. First order coefficient is zero.
M_q	$M_q' = \frac{M_q}{\frac{1}{2}\rho \ell^4 U}$	First order coefficient used in representing M as a function of q
$M_{q\eta}$	$M_{q\eta}' = \frac{M_{q\eta}}{\frac{1}{2}\rho \ell^4 U}$	First order coefficient used in representing M_q as a function of (η-1)
$M_{\dot{q}}$	$M_{\dot{q}}' = \frac{M_{\dot{q}}}{\frac{1}{2}\rho \ell^5}$	Coefficient used in representing M as a function of $\dot{q}$

$M_{q\|q\|}$	$M_{q\|q\|}' = \dfrac{M_{q\|q\|}}{\frac{1}{2}\rho \ell^5}$	Second order coefficient used in representing M as a function of q
$M_{\|q\|\delta s}$	$M_{\|q\|\delta s}' = \dfrac{M_{\|q\|\delta s}}{\frac{1}{2}\rho \ell^4 U}$	Coefficient used in representing $M_{\delta s}$ as a function q
M_{rp}	$M_{rp}' = \dfrac{M_{rp}}{\frac{1}{2}\rho \ell^5}$	Coefficient used in representing M as a function of the product rp
M_{rr}	$M_{rr}' = \dfrac{M_{rr}}{\frac{1}{2}\rho \ell^5}$	Second order coefficient used in representing M as a function of r. First order coefficient is zero
M_{vp}	$M_{vp}' = \dfrac{M_{vp}}{\frac{1}{2}\rho \ell^4}$	Coefficient used in representing M as a function of the product vp
M_{vr}	$M_{vr}' = \dfrac{M_{vr}}{\frac{1}{2}\rho \ell^4}$	Coefficient used in representing M as a function of the product vr
M_{vv}	$M_{vv}' = \dfrac{M_{vv}}{\frac{1}{2}\rho \ell^3}$	Second order coefficient used in representing M as a function of v
M_w	$M_w' = \dfrac{M_w}{\frac{1}{2}\rho \ell^3 U}$	First order coefficient used in representing M as a function of w
$M_{w\eta}$	$M_{w\eta}' = \dfrac{M_{w\eta}}{\frac{1}{2}\rho \ell^3 U}$	First order coefficient used in representing M_w as a function of $(\eta-1)$
$M_{\dot{w}}$	$M_{\dot{w}}' = \dfrac{M_{\dot{w}}}{\frac{1}{2}\rho \ell^4}$	Coefficient used in representing M as a function of $\dot{w}$
$M_{\|w\|}$	$M_{\|w\|}' = \dfrac{M_{\|w\|}}{\frac{1}{2}\rho \ell^3 U}$	First order coefficient used in representing M as a function of w; equal to zero for symmetrical function
$M_{\|w\|q}$	$M_{\|w\|q}' = \dfrac{M_{\|w\|q}}{\frac{1}{2}\rho \ell^4}$	Coefficient used in representing M_q as a function of w
$M_{w\|w\|}$	$M_{w\|w\|}' = \dfrac{M_{w\|w\|}}{\frac{1}{2}\rho \ell^3}$	Second order coefficient used in representing M as a function of w
$M_{w\|w\|\eta}$	$M_{w\|w\|\eta}' = \dfrac{M_{w\|w\|\eta}}{\frac{1}{2}\rho \ell^3}$	First order coefficient used in representing $M_{w\|w\|}$ as a function of $(\eta-1)$
M_{ww}	$M_{ww}' = \dfrac{M_{ww}}{\frac{1}{2}\rho \ell^3}$	Second order coefficient used in representing M as a function of w; equal to zero for symmetrical function
$M_{\delta b}$	$M_{\delta b}' = \dfrac{M_{\delta b}}{\frac{1}{2}\rho \ell^3 U^2}$	First order coefficient used in representing M as a function of δ_b
$M_{\delta s}$	$M_{\delta s}' = \dfrac{M_{\delta s}}{\frac{1}{2}\rho \ell^3 U^2}$	First order coefficient used in representing M as a function of δ_s
$M_{\delta s\eta}$	$M_{\delta s\eta}' = \dfrac{M_{\delta s\eta}}{\frac{1}{2}\rho \ell^3 U^2}$	First order coefficient used in representing $M_{\delta s}$ as a function of $(\eta-1)$

N	$N' = \dfrac{N}{\frac{1}{2}\rho\ell^3 U^2}$	Hydrodynamic moment component about z axis (yawing moment)
N_*	$N_*' = \dfrac{N_*}{\frac{1}{2}\rho\ell^3 U^2}$	Yawing moment when body angles (α, β) and control surface angles are zero
N_p	$N_p' = \dfrac{N_p}{\frac{1}{2}\rho\ell^4 U}$	First order coefficient used in representing N as a function of p
$N_{\dot{p}}$	$N_{\dot{p}}' = \dfrac{N_{\dot{p}}}{\frac{1}{2}\rho\ell^5}$	Coefficient used in representing N as a function of $\dot{p}$
N_{pq}	$N_{pq}' = \dfrac{N_{pq}}{\frac{1}{2}\rho\ell^5}$	Coefficient used in representing N as a function of the product pq
N_{qr}	$N_{qr}' = \dfrac{N_{qr}}{\frac{1}{2}\rho\ell^5}$	Coefficient used in representing N as a function of the product qr
N_r	$N_r' = \dfrac{N_r}{\frac{1}{2}\rho\ell^4 U}$	First order coefficient used in representing N as a function of r
$N_{r\eta}$	$N_{r\eta}' = \dfrac{N_{r\eta}}{\frac{1}{2}\rho\ell^4 U}$	First order coefficient used in representing N_r as a function of $(\eta-1)$
$N_{\dot{r}}$	$N_{\dot{r}}' = \dfrac{N_{\dot{r}}}{\frac{1}{2}\rho\ell^5}$	Coefficient used in representing N as a function of $\dot{r}$
$N_{r\|r\|}$	$N_{r\|r\|}' = \dfrac{N_{r\|r\|}}{\frac{1}{2}\rho\ell^5}$	Second order coefficient used in representing N as a function of r
$N_{\|r\|\delta r}$	$N_{\|r\|\delta r}' = \dfrac{N_{\|r\|\delta r}}{\frac{1}{2}\rho\ell^4 U}$	Coefficient used in representing $N_{\delta r}$ as a function of r
N_v	$N_v' = \dfrac{N_v}{\frac{1}{2}\rho\ell^3 U}$	First order coefficient used in representing N as a function of v
$N_{v\eta}$	$N_{v\eta}' = \dfrac{N_{v\eta}}{\frac{1}{2}\rho\ell^3 U}$	First order coefficient used in representing N_v as a function of $(\eta-1)$
$N_{\dot{v}}$	$N_{\dot{v}}' = \dfrac{N_{\dot{v}}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing N as a function of $\dot{v}$
N_{vq}	$N_{vq}' = \dfrac{N_{vq}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing N as a function of the product vq
$N_{\|v\|r}$	$N_{\|v\|r}' = \dfrac{N_{\|v\|r}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing N_r as a function of v
$N_{v\|v\|}$	$N_{v\|v\|}' = \dfrac{N_{v\|v\|}}{\frac{1}{2}\rho\ell^3}$	Second order coefficient used in representing N as a function of v
$N_{v\|v\|\eta}$	$N_{v\|v\|\eta}' = \dfrac{N_{v\|v\|\eta}}{\frac{1}{2}\rho\ell^3}$	First order coefficient used in representing $N_{v\|v\|}$ as a function of $(\eta-1)$

N_{vw}	$N_{vw}' = \frac{N_{vw}}{\frac{1}{2}\rho \ell^3}$	Coefficient used in representing N as a function of the product vw
N_{wp}	$N_{wp}' = \frac{N_{wp}}{\frac{1}{2}\rho \ell^4}$	Coefficient used in representing N as a function of the product wp
N_{wr}	$N_{wr}' = \frac{N_{wr}}{\frac{1}{2}\rho \ell^4}$	Coefficient used in representing N as a function of the product wr
$N_{\delta r}$	$N_{\delta r}' = \frac{N_{\delta r}}{\frac{1}{2}\rho \ell^3 U^2}$	First order coefficient used in representing N as a function of δ_r
$N_{\delta r \eta}$	$N_{\delta r \eta}' = \frac{N_{\delta r \eta}}{\frac{1}{2}\rho \ell^3 U^2}$	First order coefficient used in representing $N_{\delta r}$ as a function of $(\eta - 1)$
p	$p' = \frac{p\ell}{U}$	Angular velocity component about y axis relative to fluid (roll)
$\dot{p}$	$\dot{p}' = \frac{\dot{p}\ell^2}{U^2}$	Angular acceleration component about x axis relative to fluid
q	$q' = \frac{q\ell}{U}$	Angular velocity component about y axis relative to fluid (pitch)
$\dot{q}$	$\dot{q}' = \frac{\dot{q}\ell^2}{U^2}$	Angular acceleration component about y axis relative to fluid
r	$r' = \frac{r\ell}{U}$	Angular velocity component about z axis relative to fluid (yaw)
$\dot{r}$	$\dot{r}' = \frac{\dot{r}\ell^2}{U^2}$	Angular acceleration component about z axis relative to fluid
U	$U' = \frac{U}{U}$	Linear velocity of origin of body axes relative to fluid
u	$u' = \frac{u}{U}$	Component of U in direction of the x axis
$\dot{u}$	$\dot{u}' = \frac{\dot{u}\ell}{U^2}$	Time rate of change of u in direction of the x axis
u_c	$u_c' = \frac{u_c}{U}$	Command speed: steady value of ahead speed component u for a given propeller rpm when body angles (α, β) and control surface angles are zero. Sign changes with propeller reversal
v	$v' = \frac{v}{U}$	Component of U in direction of the y axis
$\dot{v}$	$\dot{v}' = \frac{\dot{v}\ell}{U^2}$	Time rate of change of v in direction of the y axis

w	$w' = \frac{w}{U}$	Component of U in direction of the z axis
$\dot{w}$	$\dot{w}' = \frac{\dot{w}\ell}{U^2}$	Time rate of change of w in direction of the z axis
W	$W' = \frac{W}{\frac{1}{2}\rho\ell^2U^2}$	Weight, including water in free flooding spaces
x	$x' = \frac{x}{\ell}$	Longitudinal body axis; also the coordinate of a point relative to the origin of body axes
x_B	$x_B' = \frac{x_B}{\ell}$	The x coordinate of CB
x_G	$x_G' = \frac{x_G}{\ell}$	The x coordinate of CG
x_o	$x_o' = \frac{x_o}{\ell}$	A coordinate of the displacement of CG relative to the origin of a set of fixed axes
X	$X' = \frac{X}{\frac{1}{2}\rho\ell^2U^2}$	Hydrodynamic force component along x axis (longitudinal, or axial, force)
X_{qq}	$X_{qq}' = \frac{X_{qq}}{\frac{1}{2}\rho\ell^4}$	Second order coefficient used in representing X as a function of q. First order coefficient is zero
X_{rp}	$X_{rp}' = \frac{X_{rp}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing X as a function of the product rp
X_{rr}	$X_{rr}' = \frac{X_{rr}}{\frac{1}{2}\rho\ell^4}$	Second order coefficient used in representing X as a function of r. First order coefficient is zero
$X_{\dot{u}}$	$X_{\dot{u}}' = \frac{X_{\dot{u}}}{\frac{1}{2}\rho\ell^3}$	Coefficient used in representing X as a function of $\dot{u}$
X_{uu}	$X_{uu}' = \frac{X_{uu}}{\frac{1}{2}\rho\ell^2}$	Second order coefficient used in representing X as a function of u in the non-propelled case. First order coefficient is zero
X_{vr}	$X_{vr}' = \frac{X_{vr}}{\frac{1}{2}\rho\ell^3}$	Coefficient used in representing X as a function of the product vr
X_{vv}	$X_{vv}' = \frac{X_{vv}}{\frac{1}{2}\rho\ell^2}$	Second order coeffient used in representing X as a function of v. First order coefficient is zero
$X_{vv\eta}$	$X_{vv\eta}' = \frac{X_{vv\eta}}{\frac{1}{2}\rho\ell^2}$	First order coefficient used in representing X_{vv} as a function of $(\eta - 1)$
X_{wq}	$X_{wq}' = \frac{X_{wq}}{\frac{1}{2}\rho\ell^3}$	Coefficient used in representing X as a function of the product wq

X_{ww}	$X_{ww}' = \frac{X_{ww}}{\frac{1}{2}\rho\ell^2}$	Second order coefficient used in representing X as a function of w. First order coefficient is zero
$X_{ww\eta}$	$X_{ww\eta}' = \frac{X_{ww\eta}}{\frac{1}{2}\rho\ell^2}$	First order coefficient used in representing X_{ww} as a function of $(\eta - 1)$
$X_{\delta b\delta b}$	$X_{\delta b\delta b}' = \frac{X_{\delta b\delta b}}{\frac{1}{2}\rho\ell^2 U^2}$	Second order coefficient used in representing X as a function of δ_b. First order coefficient is zero
$X_{\delta r\delta r}$	$X_{\delta r\delta r}' = \frac{X_{\delta r\delta r}}{\frac{1}{2}\rho\ell^2 U^2}$	Second order coefficient used in representing X as a function of δ_r. First order coefficient is zero
$X_{\delta r\delta r\eta}$	$X_{\delta r\delta r\eta}' = \frac{X_{\delta r\delta r\eta}}{\frac{1}{2}\rho\ell^2 U^2}$	First order coefficient used in representing $X_{\delta r\delta r}$ as a function of $(\eta - 1)$
$X_{\delta s\delta s}$	$X_{\delta s\delta s}' = \frac{X_{\delta s\delta s}}{\frac{1}{2}\rho\ell^2 U^2}$	Second order coefficient used in representing X as a function of δ_s. First order coefficient is zero
$X_{\delta s\delta s\eta}$	$X_{\delta s\delta s\eta}' = \frac{X_{\delta s\delta s\eta}}{\frac{1}{2}\rho\ell^2 U^2}$	First order coefficient used in representing $X_{\delta s\delta s}$ as a function of $(\eta - 1)$
y	$y' = \frac{y}{\ell}$	Lateral body axis; also the coordinate of a point relative to the origin of body axes
y_B	$y_B' = \frac{y_B}{\ell}$	The y coordinate of CB
y_G	$y_G' = \frac{y_G}{\ell}$	The y coordinate of CG
y_o	$y_o' = \frac{y_o}{\ell}$	A coordinate of the displacement of CG relative to the origin of a set of fixed axes
Y	$Y' = \frac{Y}{\frac{1}{2}\rho\ell^2 U^2}$	Hydrodynamic force component along y axis (lateral force)
Y_*	$Y_*' = \frac{Y}{\frac{1}{2}\rho\ell^2 U^2}$	Lateral force when body angles (α, β) and control surface angles are zero
Y_p	$Y_p' = \frac{Y_p}{\frac{1}{2}\rho\ell^3 U}$	First order coefficient used in representing Y as a function of p
$Y_{\dot{p}}$	$Y_{\dot{p}}' = \frac{Y_{\dot{p}}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing Y as a function of $\dot{p}$
$Y_{p\|p\|}$	$Y_{p\|p\|}' = \frac{Y_{p\|p\|}}{\frac{1}{2}\rho\ell^4}$	Second order coefficient used in representing Y as a function of p

Y_{pq}	$Y_{pq}' = \frac{Y_{pq}}{\frac{1}{2}\rho \ell^4}$	Coefficient used in representing Y as a function of the product pq
Y_{qr}	$Y_{qr}' = \frac{Y_{qr}}{\frac{1}{2}\rho \ell^4}$	Coefficient used in representing Y as a function of the product qr
Y_r	$Y_r' = \frac{Y_r}{\frac{1}{2}\rho \ell^3 U}$	First order coefficient used in representing Y as a function of r
$Y_{r\eta}$	$Y_{r\eta}' = \frac{Y_{r\eta}}{\frac{1}{2}\rho \ell^3 U}$	First order coefficient used in representing Y_r as a function of $(\eta-1)$
$Y_{\dot{r}}$	$Y_{\dot{r}}' = \frac{Y_{\dot{r}}}{\frac{1}{2}\rho \ell^4}$	Coefficient used in representing Y as a function of $\dot{r}$
$Y_{\|r\|\delta r}$	$Y_{\|r\|\delta r}' = \frac{Y_{\|r\|\delta r}}{\frac{1}{2}\rho \ell^3 U}$	Coefficient used in representing $Y_{\delta r}$ as a function of r
Y_v	$Y_v' = \frac{Y_v}{\frac{1}{2}\rho \ell^2 U}$	First order coefficient used in representing Y as a function of v
$Y_{v\eta}$	$Y_{v\eta}' = \frac{Y_{v\eta}}{\frac{1}{2}\rho \ell^2 U}$	First order coefficient used in representing Y_v as a function of $(\eta-1)$
$Y_{\dot{v}}$	$Y_{\dot{v}}' = \frac{Y_{\dot{v}}}{\frac{1}{2}\rho \ell^3}$	Coefficient used in representing Y as a function of $\dot{v}$
Y_{vq}	$Y_{vq}' = \frac{Y_{vq}}{\frac{1}{2}\rho \ell^3}$	Coefficient used in representing Y as a function of the product vq
$Y_{v\|r\|}$	$Y_{v\|r\|}' = \frac{Y_{v\|r\|}}{\frac{1}{2}\rho \ell^3}$	Coefficient used in representing Y_v as a function of r
$Y_{v\|v\|}$	$Y_{v\|v\|}' = \frac{Y_{v\|v\|}}{\frac{1}{2}\rho \ell^2}$	Second order coefficient used in representing Y as a function of v
$Y_{v\|v\|\eta}$	$Y_{v\|v\|\eta}' = \frac{Y_{v\|v\|\eta}}{\frac{1}{2}\rho \ell^2}$	First order coefficient used in representing $Y_{v\|v\|}$ as a function of $(\eta-1)$
Y_{vw}	$Y_{vw}' = \frac{Y_{vw}}{\frac{1}{2}\rho \ell^2}$	Coefficient used in representing Y as a function of the product vw
Y_{wp}	$Y_{wp}' = \frac{Y_{wp}}{\frac{1}{2}\rho \ell^3}$	Coefficient used in representing Y as a function of the product wp
Y_{wr}	$Y_{wr}' = \frac{Y_{wr}}{\frac{1}{2}\rho \ell^3}$	Coefficient used in representing Y as a function of the product wr
$Y_{\delta r}$	$Y_{\delta r}' = \frac{Y_{\delta r}}{\frac{1}{2}\rho \ell^2 U^2}$	First order coefficient used in representing Y as a function of δr
$Y_{\delta r\eta}$	$Y_{\delta r\eta}' = \frac{Y_{\delta r\eta}}{\frac{1}{2}\rho \ell^2 U^2}$	First order coefficient used in representing $Y_{\delta r}$ as a function of $(\eta-1)$

z	$z' = \frac{z}{\ell}$	Normal body axis; also the coordinate of a point relative to the origin of body axes
z_B	$z_B' = \frac{z_B}{\ell}$	The z coordinate of CB
z_G	$z_G' = \frac{z_G}{\ell}$	The z coordinate of CG
z_o	$z_o' = \frac{z_o}{\ell}$	A coordinate of the displacement of CG relative to the origin of a set of fixed axes
Z	$Z' = \frac{Z}{\frac{1}{2}\rho\ell^2U^2}$	Hydrodynamic force component along z axis (normal force)
Z_*	$Z_*' = \frac{Z_*}{\frac{1}{2}\rho\ell^2U^2}$	Normal force when body angles (α, β) and control surface angles are zero
Z_{pp}	$Z_{pp}' = \frac{Z_{pp}}{\frac{1}{2}\rho\ell^4}$	Second order coefficient used in representing Z as a function of p. First order coefficient is zero
Z_q	$Z_q' = \frac{Z_q}{\frac{1}{2}\rho\ell^3U}$	First order coefficient used in representing Z as a function of q
$Z_{q\eta}$	$Z_{q\eta}' = \frac{Z_{q\eta}}{\frac{1}{2}\rho\ell^3U}$	First order coefficient used in representing Z_q as a function of $(\eta-1)$
$Z_{\dot{q}}$	$Z_{\dot{q}}' = \frac{Z_{\dot{q}}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing Z as a function of $\dot{q}$
$Z_{\|q\|\delta s}$	$Z_{\|q\|\delta s}' = \frac{Z_{\|q\|\delta s}}{\frac{1}{2}\rho\ell^3U}$	Coefficient used in representing $Z_{\delta s}$ as a function of q
Z_{rp}	$Z_{rp}' = \frac{Z_{rp}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing Z as a function of the product rp
Z_{rr}	$Z_{rr}' = \frac{Z_{rr}}{\frac{1}{2}\rho\ell^4}$	Second order coefficient used in representing Z as a function of r. First order coefficient is zero
Z_w	$Z_w' = \frac{Z_w}{\frac{1}{2}\rho\ell^2U}$	First order coefficient used in representing Z as a function of w
$Z_{w\eta}$	$Z_{w\eta}' = \frac{Z_{w\eta}}{\frac{1}{2}\rho\ell^2U}$	First order coefficient used in representing Z_w as a function of $(\eta-1)$
$Z_{\dot{w}}$	$Z_{\dot{w}}' = \frac{Z_{\dot{w}}}{\frac{1}{2}\rho\ell^3}$	Coefficient used in representing Z as a function of $\dot{w}$
$Z_{\|w\|}$	$Z_{\|w\|}' = \frac{Z_{\|w\|}}{\frac{1}{2}\rho\ell^2U}$	First order coefficient used in representing Z as a function of w; equal to zero for symmetrical function
$Z_{w\|q\|}$	$Z_{w\|q\|}' = \frac{Z_{w\|q\|}}{\frac{1}{2}\rho\ell^3}$	Coefficient used in representing Z_w as a function of q

$Z_{w\|w\|}$	$Z_{w\|w\|}' = \frac{Z_{w\|w\|}}{\frac{1}{2}\rho \ell^2}$	Second order coefficient used in representing Z as a function of w
$Z_{w\|w\|\eta}$	$Z_{w\|w\|\eta}' = \frac{Z_{w\|w\|\eta}}{\frac{1}{2}\rho \ell^2}$	First order coefficient used in representing $Z_{w\|w\|}$ as a function of $(\eta-1)$
Z_{ww}	$Z_{ww}' = \frac{Z_{ww}}{\frac{1}{2}\rho \ell^2}$	Second order coefficient used in representing Z as a function of w; equal to zero for symmetrical function
$Z_{\delta b}$	$Z_{\delta b}' = \frac{Z_{\delta b}}{\frac{1}{2}\rho \ell^2 U^2}$	First order coefficient used in representing Z as a function of δ_b
$Z_{\delta s}$	$Z_{\delta s}' = \frac{Z_{\delta s}}{\frac{1}{2}\rho \ell^2 U^2}$	First order coefficient used in representing Z as a function of δ_s
$Z_{\delta s\eta}$	$Z_{\delta s\eta}' = \frac{Z_{\delta s\eta}}{\frac{1}{2}\rho \ell^2 U^2}$	First order coefficient used in representing $Z_{\delta s}$ as a function of $(\eta-1)$
α		Angle of attack
β		Angle of drift
δ_b		Deflection of bowplane or sailplane
δ_r		Deflection of rudder
δ_s		Deflection of sternplane
η		The ratio $\frac{u_c}{U}$
θ		Angle of pitch
ψ		Angle of yaw
ϕ		Angle of roll
a_i, b_i, c_i		Sets of constants used in the representation of propeller thrust in the axial equation

Naval Training Device Center, Orlando, Fla.

NAVTRADEVCEN 68-C-0050-2 UNCLASSIFIED

AD________

ADVANCED SUBMARINE SYSTEMS PROGRAMING.
FINAL REPORT. JULY 1969,
373p, 23 illus., 25 tables, 4 refs.

This report covers the FORTRAN programing for two submarine simulations using the Naval Ship Research and Development Center standard equations of motion. Programs are given to extend the main simulation programs for research. A small computer simulation and a method of generating random ocean waves is included. The report covers program descriptions, FORTRAN listings, flow charts, input decks and output sheets.

DESCRIPTORS

Simulators
Digital Simulation
Training Devices
Submarine Simulators
Equations of Motion
Mathematical Models
Numerical Integration
Ocean Waves

Goodyear Aerospace Corp.
Groves, B.R. and others
N61339-68-C-0050

Naval Training Device Center, Orlando, Fla.

NAVTRADEVCEN 68-C-0050-2 UNCLASSIFIED

AD________

ADVANCED SUBMARINE SYSTEMS PROGRAMING.
FINAL REPORT. JULY 1969,
373p, 23 illus., 25 tables, 4 refs.

This report covers the FORTRAN programing for two submarine simulations using the Naval Ship Research and Development Center standard equations of motion. Programs are given to extend the main simulation programs for research. A small computer simulation and a method of generating random ocean waves is included. The report covers program descriptions, FORTRAN listings, flow charts, input decks and output sheets.

DESCRIPTORS

Simulators
Digital Simulation
Training Devices
Submarine Simulators
Equations of Motion
Mathematical Models
Numerical Integration
Ocean Waves

Goodyear Aerospace Corp.
Groves, B.R. and others
N61339-68-C-0050

Naval Training Device Center, Orlando, Fla.

NAVTRADEVCEN 68-C-0050-2 UNCLASSIFIED

AD________

ADVANCED SUBMARINE SYSTEMS PROGRAMING.
FINAL REPORT. JULY 1969,
373p, 23 illus., 25 tables, 4 refs.

This report covers the FORTRAN programing for two submarine simulations using the Naval Ship Research and Development Center standard equations of motion. Programs are given to extend the main simulation programs for research. A small computer simulation and a method of generating random ocean waves is included. The report covers program descriptions, FORTRAN listings, flow charts, input decks and output sheets.

DESCRIPTORS

Simulators
Digital Simulation
Training Devices
Submarine Simulators
Equations of Motion
Mathematical Models
Numerical Integration
Ocean Waves

Goodyear Aerospace Corp.
Groves, B.R. and others
N61339-68-C-0050

Naval Training Device Center, Orlando, Fla.

NAVTRADEVCEN 68-C-0050-2 UNCLASSIFIED

AD________

ADVANCED SUBMARINE SYSTEMS PROGRAMING.
FINAL REPORT. JULY 1969,
373p, 23 illus., 25 tables, 4 refs.

This report covers the FORTRAN programing for two submarine simulations using the Naval Ship Research and Development Center standard equations of motion. Programs are given to extend the main simulation programs for research. A small computer simulation and a method of generating random ocean waves is included. The report covers program descriptions, FORTRAN listings, flow charts, input decks and output sheets.

DESCRIPTORS

Simulators
Digital Simulation
Training Devices
Submarine Simulators
Equations of Motion
Mathematical Models
Numerical Integration
Ocean Waves

Goodyear Aerospace Corp.
Groves, B.R. and others
N61339-68-C-0050

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Goodyear Aerospace Corp., Akron, Ohio	2a. REPORT SECURITY CLASSIFICATION Unclassified
	2b. GROUP

3. REPORT TITLE

ADVANCED SUBMARINE SYSTEMS PROGRAMING

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Final Report

5. AUTHOR(S) (First name, middle initial, last name)

Brenton R. Groves, Ph.D.
James T. Dorsey
Dennis Tucker

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
November 1969	373	4

8a. CONTRACT OR GRANT NO. N61339-68-C-0050	9a. ORIGINATOR'S REPORT NUMBER(S) GER-14449
b. PROJECT NO. 7843	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	NAVTRADEVCEN 68-C-0050-2

10. DISTRIBUTION STATEMENT

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of Commanding Officer, Naval Training Device Center, Orlando, Florida 32813 (Code ~~424~~ 423)

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	U.S. Naval Training Device Center Orlando, Fla.

13. ABSTRACT

This programing report is the result of a study leading to the determination of the optimum sets of equations of motion to be used with two general types of submarine control trainers. The starting point was the Naval Ship Research and Development Center standard equations of motion for submarine simulators.

Two complete submarine simulation programs using these equations are given; one for six-degrees-of-freedom and one for the longitudinal three-degrees-of-freedom. A number of programs are included to assist the researcher in interpreting the results of the main simulation program. A compact submarine simulation program for use with a small computer is given and a method of generating random ocean wave amplitudes is outlined along with its program.

This report describes the programs, including listing in FORTRAN, flow charts, input decks, and typical output sheets, but does not tell how they are to be used. It should be used in conjunction with Advanced Submarine Systems Equation Study, NAVTRADEVCEN 68-C-0050-1 which describes the work performed under this study.

DD FORM 1 NOV 65 1473

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Simulators Training Devices Digital Simulation Submarine Simulators Equations of Motion Mathematical Models Numerical Integration Ocean Waves						